Methods and Technologies for Using Body Motion for Real-Time Musical Interaction

Ph.D. thesis

Ståle Andreas van Dorp Skogstad

Thursday 26th September, 2013

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 1453

ISSN 1501-7710

Cover: Inger Sandved Anfinsen.
Printed in Norway: AIT Oslo AS.

Produced in co-operation with Akademika Publishing.
The thesis is produced by Akademika Publishing merely in connection with the thesis defence. Kindly direct all inquiries regarding the thesis to the copyright holder or the unit which grants the doctorate.

Abstract

There are several strong indications for a profound connection between *musical sound* and *body motion*. Musical *embodiment*, meaning that our bodies play an important role in how we experience and understand music, has become a well accepted concept in music cognition. Today there are increasing numbers of new *motion capture* (MoCap) technologies that enable us to incorporate the paradigm of musical embodiment into *computer music*. This thesis focuses on some of the challenges involved in designing such systems. That is, how can we design *digital musical instruments* that utilize MoCap systems to map *motion* to *sound*?

The first challenge encountered when wanting to use body motion for musical interaction is to find appropriate MoCap systems. Given the wide availability of different systems, it has been important to investigate the strengths and weaknesses of such technologies. This thesis includes evaluations of two of the technologies available: an optical marker-based system known as OptiTrack V100:R2; and an inertial sensor-based system known as the Xsens MVN suit.

Secondly, to make good use of the raw MoCap data from the above technologies, it is often necessary to process them in different ways. This thesis presents a review and suggestions towards best practices for processing MoCap data in real time. As a result, several novel methods and filters that are applicable for processing MoCap data for real-time musical interaction are presented in this thesis. The most reasonable processing approach was found to be utilizing digital filters that are designed and evaluated in the frequency domain. To determine the frequency content of MoCap data, a frequency analysis method has been developed. An experiment that was carried out to determine the typical frequency content of free hand motion is also presented. Most remarkably, it has been necessary to design filters with low time delay, which is an important feature for real-time musical interaction. To be able to design such filters, it was necessary to develop an alternative filter design method. The resulting noise filters and differentiators are more low-delay optimal than than those produced by the established filter design methods.

Finally, the interdisciplinary challenge of making good couplings between motion and sound has been targeted through the Dance Jockey project. During this project, a system was developed that has enabled the use of a full-body inertial motion capture suit, the Xsens MVN suit, in music/dance performances. To my knowledge, this is one of the first attempts to use a full body MoCap suit for musical interaction, and the presented system has demonstrated several hands-on solutions for how such data can be used to control sonic and musical features. The system has been used in several public performances, and the conceptual motivation, development details and experience of using the system are presented.

Preface

The thesis is written for the Faculty of Mathematics and Natural Sciences at the University of Oslo for the degree of Philosophiae Doctor (Ph.D.). The work has been funded by the Research Council of Norway, through the research project Sensing Music-Related Actions (SMA) with project number 183180. The research was conducted between 2008 and 2012, under the supervision of Mats Høvin, and co-supervision of Alexander Refsum Jensenius, Rolf Inge Godøy, Jim Tørresen and Sverre Holm. The work has been done within the interdisciplinary research group fourMs (Music, Mind, Motion, Machines), involving researchers from the Department of Musicology and the Robotics and Intelligent Systems research group (ROBIN) at the Department of Informatics.

Acknowledgments

I have many people to thank for help and support during the period that I have been working on this thesis. First and foremost, I am grateful to my supervisors who have provided invaluable advice and support throughout the entire Ph.D. project. Mats Høvin has pushed me forward and has been an important main supervisor, both professionally and personally. Alexander Refsum Jensenius has been the supervisor with the broadest knowledge base, which has been crucial for finishing this thesis. Additionally, Rolf Inge Godøy and Jim Tørresen have provided me with important knowledge and resources during the work of this thesis. Since a large part of my research has consisted of details in digital signal processing, I'm very grateful for the additional supervision I received from Sverre Holm.

Next, I need to thank my good colleagues in the fourMs and ROBIN research groups in Oslo. Many discussions with fellow Ph.D. student Kristian Nymoen have been an essential part of surviving the never ending challenges that a Ph.D. presents. I would also like to thank Yago de Quay which has been an essential partner during the Dance Jockey project. Additionally, I need to thank *Gordon* who has been a big help with his language skills. I also want to thank the rest of my colleagues, *Arjun*, *Arve*, *Dirk*, *Kyrre*, *Ripon*, *Yngve*, *Kim*, *Alexander*, *Markus*, *Simen* and *Eivind*. It has been a delightful working atmosphere.

My hunger for knowledge has been an important motivation for this research. This desire for knowledge would not have been were it not for some great teachers and friends. I would also like to thank my family for all their support and their keen interest in seeing me finish my Ph.D. During the work of this thesis, I lost my two most important supporters, who in many ways are the reason I am who I am. Thank you! Finally, I want to give a special, warm thank you to Tine, who had her share of the price of me doing this Ph.D. I hope I can make it up to you!

Contents

Chapter 1

Introduction

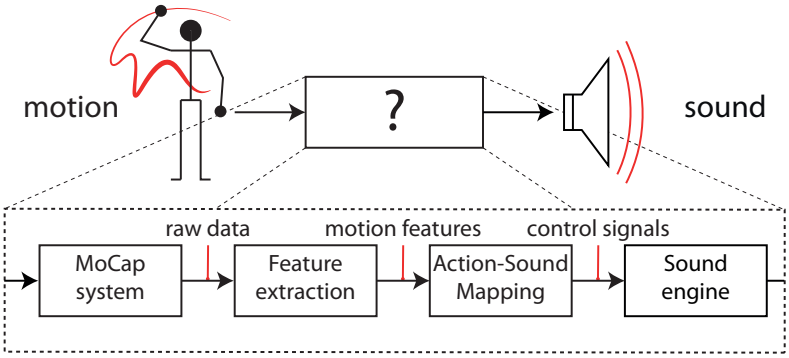


Figure 1.1: A data flow diagram which illustrates how human motion can be transferred into musical expression. The general research question of this thesis is how motion can be used to control sonic and musical features, illustrated by the question mark.

The research presented in this dissertation is focused on technologies and methods for the use of motion capture systems in real-time musical interaction. The underlying goal is to make systems that "transform" human body motion into musical expressions. Figure 1.1 gives a data flow illustration of how such a system can be built. First of all, we need a *motion capture system* (MoCap) that can track our motion in real time. Then we need to extract some *motion features* from the raw motion capture data that are suitable to *map* to *control signals* for the sound engine, and finally, the *sound engine* is responsible for translating the control signals into musical features or sonic events. As a result, the system maps *motion* to *sound.* Each of these parts involves several different challenges, and some of them are targeted in this thesis. Before I go into the details of the aims and objectives of this thesis, let us first consider the motivation for pursuing this topic.

1.1 Motivation

What came first, music or motion?

Most people will agree that *music* and *motion* have some profound connections. Not only is music a result of motion when we play musical instruments, but listening to music can often lead to spontaneous motion, e.g. tapping our fingers and feet, or even joyful dancing. *Spontaneous* may be a correct term since research suggests that infants have a predisposition toward such embodied entrainment [70]. In other words, we may have been born with a predisposition toward *moving to music*. Whatever the reasons are for this spontaneous need to rock to music, music and dance are often experienced as joyful and an important part of many social and cultural events. Additionally, a recent review of the literature gives support to the claim that music has a positive influence on our health [6].

There are several reasons that explain why music can be an important part of life, and this may in part be a result of so-called musical embodiment, i.e. experience of music is intimately linked with the experience of our body [17]. Recent studies suggest also that our experience with action-sound couplings, based on relationships between actions, objects, and the resultant sounds, guide the way we think about both actions and sounds [19, 27]. In this way, we can say that music is *multimodal,* i.e., it is not only communicated through the auditory modality, since when listening to music we also form mental images that are more related to other modalities, e.g. sensations of sound-producing actions like smooth, hard, jerky, slow, etc. [18]. Today, there are several motion capture technologies available that allow us to study the intriguing relationship between music and motion in a quantitative way [40, 5, 56]. Yet such technologies do not only allow us to study how we move to music; we could take it even further and use these technologies to make *new music*. This is precisely the focus of this thesis.

As you might suspect, the cumbersome course of using arbitrary body motion to play a melodic tune, will probably never surpass the simplicity of using the much more straightforward path of buttons, knobs and interfaces like the piano keyboard. On the other hand, such motion interfaces can provide alternative ways of making music that are closer to the paradigm of musical embodiment. This can be beneficial for instrument design, since our body plays an important part in how we experience and understand music. Imagine a virtual motion instrument that enables you to express yourself, without the need for complex motoric skills and years of practice. Such alternative musical instruments may also be beneficial for disabled people who are not able to play traditional instruments [62]. Yet, this may be beneficial not only for the instrumental performer, but also for the spectator.

Electronic music, i.e. music made by computers and sound synthesizers, has clearly given rise to a vast set of new sonic possibilities. However, it is often commented that the genre typically lacks a physical presence during live performance [3]. This may simply be a manifestation of the genre, i.e. they use computers and not acoustic instruments that require specific physical actions on stage. Nevertheless, this has been an additional motivation for investigating how new motion capture technologies can be used for exploring new musical expression, both privately and for an audience, with a greater physical involvement and presence.

1.2 The Dance Jockey project

Figure 1.2: A Dance Jockey performance at Mostra UP in Porto, Portugal. Notice the orange sensors on different body parts which are parts of the MoCap suit used.

During this PhD project, Yago de Quay and I have worked with the *Dance Jockey Project.* The main goal was to make a musical performance piece based on *full body motion data*, inspired by the above ideas and motivation. To my knowledge, this was the first time someone had attempted to use a full body MoCap suit, i.e. a wearable suit that tracks the motion of the main limbs of the whole body, for real-time musical performance. Developing the Dance Jockey system involved several challenges. First of all, it consisted of various technical details, e.g. incorporation of the MoCap system and development of the necessary *real-time* software and algorithms. *Real-time* is an important keyword, since low latency is seen as an important property for achieving intimate control in musical applications [65]. Processing MoCap data with low delay is therefore a significant focus of this thesis. Secondly, there were also high-level design challenges, as opposed to low-level implementation details, that needed to be addressed, e.g., how do we create good mappings between motion and sound? Such questions and challenges have been targeted in this thesis. Before formulating these questions and challenges into the aims and objectives of this thesis, let us first briefly consider the limitations.

1.3 Interdisciplinary and limitations

The research that is presented in this thesis covers several different fields, e.g. human computer interaction, motion capture technologies, digital signal processing, multi-objective optimization and heuristic search. However, there are several more important fields and challenges which would have been relevant to study, e.g. sound synthesis and music cognition. Due to the limited time and resources, it has been necessary to select some priorities. Given my background in computer science and technology, it has been natural to concentrate on the technical challenges. In other words, this thesis is focused on the technical side of the targeted challenges. Let us now consider the research objectives of this thesis.

1.4 Research aims and objectives

The main research objective of this thesis is to:

> *develop methods and technologies for using body motion for real-time musical interaction*

This objective can further be divided into the following sub-objectives:

- *Evaluate* different motion capture technologies for real-time musical interaction.
- *Investigate* how *full body* motion capture data can be used for musical performance.
- *Review* and *study* best practices for filtering MoCap data for real-time applications.

1.5 Thesis outline

This thesis is a collection of papers and thus the seven included research papers constitute the main research contribution of the thesis. Given the brevity of the research papers, some additional details and background are included in the following chapters. Figure 1.3 shows how these chapters are related to our challenge, and the outline is as follows.

- **Chapter 2**: *Digital musical instruments in a human-computer interaction view*
  In this chapter, inspired by the field of human-computer interaction (HCI), some aspects of the targeted design challenge are presented which I deem important when designing good action-sound mappings. The ideas and concepts that are presented in this section have been the main motivation behind the work I did in the *Dance Jockey project*.
- **Chapter 3**: *Motion capture*
  The first step in our challenge is to capture the wanted body motion. This chapter presents a brief overview and the essential challenges of MoCap technologies, with some additional details and considerations about the MoCap systems which have been used in this thesis.
- **Chapter 4:** *Filtering MoCap data*.
  To make good use of the MoCap data, it is often necessary to process it in different ways. In this chapter I first give some background to digital filter design and continue by discussing best practices for *noise filtering* and *differentiating* of MoCap data. Since the filters are intended for real-time applications, an important focus is on designing such filters with low delay. To be able to explore and design optimal low-delay filters, it was necessary to develop an alternative filter design method. This is the most detailed chapter and gives additional information and background to the results given in Papers V, VI and VII, which are significant parts of the contribution of this thesis.

I then continue by presenting an overview of the contents of the research papers, as well as individual motivations and abstracts for each paper in **Chapter 5.** This chapter also lists the Dance Jockey performances that have been performed and some software that has been made

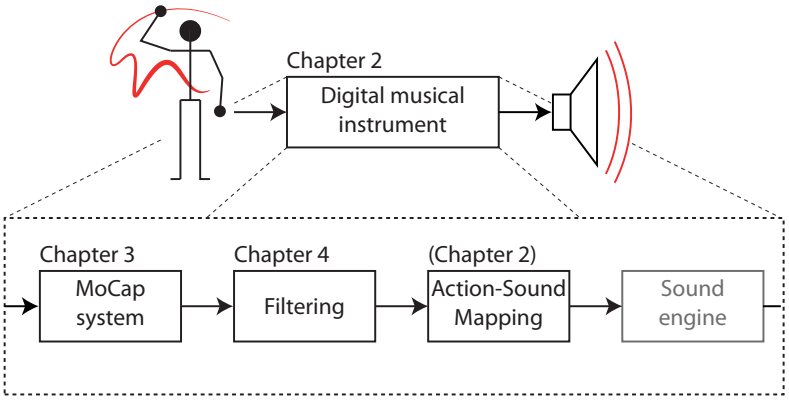


Figure 1.3: Organization of the background chapters. Notice that the sound engine is not a focused part of this thesis.

available to others. Subsequently, **Chapter 6** presents a summary of this thesis and proposes future work. Finally, the seven research papers are included at the end of the thesis. Additional details on some of the proposed work are given in the **Appendix**.

The reader of this thesis is not assumed to have any special knowledge of the terminology and methods used in this thesis. For this reason, the terminology, technologies and methods presented in chapters 2, 3 and 4 will be presented in such a way that they are accessible without expert knowledge.

Chapter 2

Digital musical instruments in a human-computer interaction view

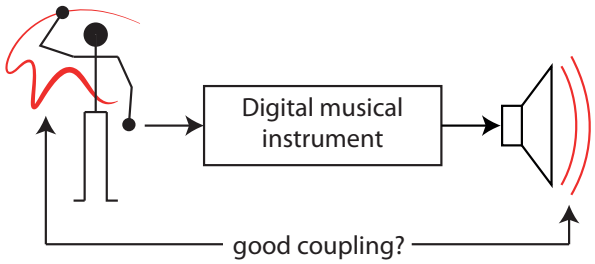


Figure 2.1: How can we design good couplings between motion and sound?

In this chapter I will discuss some high-level design aspects of interfaces for musical expression and relate these to existing literature. Inspired by the world of *human-computer interaction* (HCI), I present a conceptual model that I believe is important for understanding a basic challenge of the interdisciplinary complexity in musical instrument design. This model suggests that interface design should be guided by our perceptual and cognitive constraints. I raise the question of what the main elements of intuitive control of music are and, based on the conceptual model, I propose a basic design rule, including a list of accompanying concepts, which I deem important when forging a good coupling between action and sound.

2.1 Introduction

The field of *human-computer interaction* incorporates many challenges regarding the design of the interaction between users and computers. HCI is often regarded as the intersection of computer science, behavioral sciences, design and several other fields of study. The scope of this chapter is not to review the whole field but to consider the challenge that is investigated in this thesis in an HCI view and take inspiration from some of the established ideas.

The design challenge of this thesis can be called a *digital musical instrument* (DMI). More specifically, I am interested in instruments used to transform body motion into musical expressions, i.e. sound or musical features. It is evident that today's computers can make sound,

and with digital *controllers* and real-time audio software, we can control sound in real time. Consequently, we can perform music with *digital musical instruments*. Every sensor that can sense some aspects of the physical world can be used as a controller, as attested to by the many examples found in the literature [36].

Since our problem is related to HCI, it is natural to turn to this field when wanting to analyze and evaluate a DMI. However, as Wanderley et al. claim, "*Interactive computer music can be seen as a highly specialized field of HCI*" [60]. HCI theory is not necessarily applicable when designing a DMI, since the challenges of a DMI design are not identical to those of an HCI design. With computers we want to work as fast and efficiently as possible, while the goal with a DMI design is more complex than to obtain efficient and ergonomic properties [23, 29, 38]. An additional aspect is how the audience perceives the DMI design in a performance setting. Not only are the outputted sounds important, but also how the sounds relate to the performers' actions on stage [3].

Jacob claims that a fundamental goal of research in human-computer interaction is to increase the useful bandwidth of interfaces [24]. This sounds like a reasonable goal for a DMI design, since increasing the communications flow between the user and the instrument should increase the connection with the instrument or the *control intimacy* [38]. In the following I argue that the design should take advantage of our so-called *ecological knowledge* of sound, to make a more intuitive DMI. This is the idea I pursue in this chapter.

In the next section I discuss what I see as the higher-level design constraints of a DMI. In section 2.3 I continue by presenting a conceptual model of a DMI design, including a design goal. Subsequently, in section 2.4, I give an example from HCI to illustrate the concept of this design goal. In section 2.5 I continue by listing some concepts that I argue can be valuable when designing DMI. Finally, in Section 2.6, I give a discussion of this chapter.

2.2 DMI design constraints and ecological knowledge

A relevant question when designing a DMI is to consider the general design constraints. We can start by arguing that the user's ability to interact with a device is constrained by the nature of human attention, cognition, perceptual-motor skills and abilities [1], whereas a DMI design is limited by the technology used. At first it is natural to regard our body's action capabilities as the major constraint. However, one should not underestimate the complexities of motor control; just consider our vocal apparatus with its around 40 muscles and very rich output possibilities. Such control possibilities, combined with the emerging range of new sensor and digital signal processing technology, should allow us to make highly advanced DMIs. At the same time, a too complicated DMI can overload our perceptual apparatus and make it difficult to master and enjoy. The current range of available and popular instruments may provide an idea for what a good balance between learnability and complexity is [33]. In other words, while a good instrument is clearly not only about user-friendliness, it should be reasonable to regard a too *complicated* and *non-intuitive* DMI design as not beneficial in terms either of its expressivity or its mastering potential (learnability).

An advantage of acoustic instruments is that they follow the laws of physics. These laws, or constraints, determine the instrument's behavior which is perceived with our many different

senses [13]. In other words, our perception has many sources of sensory information to build a more complex model of a sonic event. Dealing with the physical world over the course of time has made us experts at negotiating these constraints. We can more or less predict how it will sound if we do something with a physical object [14, 16, 8]. In this way, we can say that the *control space* of the object has an *intuitive* connection to the *output space*. We have an idea of how to make that wanted sound since we have a deeper knowledge and understanding of how the instrument works. I argue that a DMI design can benefit from mimicking some of these constraints, so that it can benefit from our *ecological knowledge*, meaning accumulated knowledge of sound and sound-making and how they are related to the physical world. Granted that this is the case, we may now, through a conceptual model, define the terms *control space* and *output space*.

2.3 A simple conceptual metaphor for DMI

In HCI a conceptual metaphor is often used as a high-level description of how a system works [51]. The model should be an abstraction that outlines the most important system properties and shows how these are related. It is possible to make these models highly complicated by trying to incorporate every property in detail. However, the goal here is to make a simple model that will serve a specific purpose. Inspired by a model from HCI literature [21], we can define the following conceptual model for DMI.

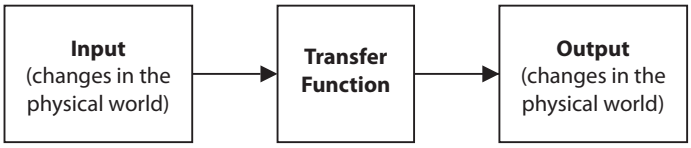


Figure 2.2: Conceptual model 1 - The technical model

Conceptual model 1*:* A DMI is a device that connects a physical change in the world to another physical change in the world through a transfer function. The first is seen as the input while the latter is seen as the output. (Figure 2.2)

The different parts of the model can be further defined as the following.

- *The input* possibilities are endless but we will mostly think about input initiated by users, as what we call *actions*. A term known from literature is *musical gestures*, but since this term includes more than the controlling actions per se, I choose to use the term action, meaning intended motion that is meant to make or manipulate sound [28]. An added importance for DMI in a performance setting is what the audience perceives from these actions [39, 11, 57].

- *The transfer function* is the core of the DMI that *maps* input to output and is often referred to as the *mapping problem*. Several publications discuss this important challenge but focus mainly on the mapping between the input signal and sound, with less focus on the perceptual and cognitive aspects of the *whole design*, mostly also omitting haptic feedback from their mapping model [23, 59, 2, 9].

- *The output* includes everything that comes out of this device, such as sound, tactile vibration and all the other output that can be sensed by the performer and audience. Digital controllers often lack the physical response and haptic feedback that acoustic instruments give. This must therefore be implemented in the design as an extra output attribute, and is referred to as *tactile*, *force* or *haptic* feedback [58, 35].

Notice that haptic feedback is mostly a concern for the control aspects of the device, i.e. how the instrument is tactually perceived by the performer, and not directly relevant for the intended output sound. How the instrument is *perceived* can in many cases be more important than how it works. We shall therefore now transform the *technical model* into the following *perceptual model*.

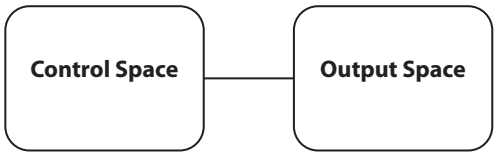


Figure 2.3: Conceptual model 2 - The perceptual model

Conceptual model 2: A DMI is a device that offers a control space and connects it to an output space. (Figure 2.3)

- The *Control Space* is how the performer experiences the DMI as a control interface. This includes the haptic feedback. The audience may also perceive some of the aspects of the control space, but not necessarily to the same degree.
- The *Output Space* is how the DMI is experienced as a sound generator, perceived by both performer and audience. We can loosely say that it consists of the intended output of the system.

We could have included more details in the above definitions; however, as mentioned at the beginning of this section, this model is meant to serve a specific purpose. The main point of the above conceptual models is to incorporate the whole transfer function, including every perceived element of the DMI. As stated by Hinckley et al. [21], an input device can not be studied without examining the intended output, for the obvious reason that the output is a fundamental part of the interaction. Likewise, I argue that a DMI can not be analyzed as a musical instrument without taking account of the whole conceptual model. Others have also stated similar ideas on DMI related to the mapping problem [23].

With the *conceptual model 2*, which is based on the *conceptual model 1*, I propose the following simple design goal: *The control space should, to some degree, match the output space.* Let us consider an example from the field of HCI to explain and illustrate the concept behind this design goal.

2.4 Moving a position marker on the graphical screen

Figure 2.4: The *mouse* is better perceptually understood as a position marker mover than the *pointing stick* on a laptop. This is because the match between control space and output space is better.

A joystick may be regarded as a two-dimensional force sensor and has often been used as a position marker mover device, e.g. a pointing stick on a laptop. How well suited is this device for the task of moving a marker on the graphical screen? Intuitively, some will think it is not optimal – but why? We can claim that the control space does not match the output space well, since the joystick is better perceptually understood as a two degrees of freedom *force sensor* than a position marker mover. You will probably with little effort learn that to move the arrow you need to push the stick in the appropriate direction. However, as you may have experienced, accurate control of speed and moving the marker to the target position can be difficult and frustrating.

Balakrishnan et al. list in [4] several reasons why a mouse works well with the graphical screen. You move the mouse and get a direct corresponding movement on the screen. The match between control and output space is better than the joystick example. To achieve this direct bond is clearly important; however, with DMI it may be difficult to achieve because the qualities of sound, like timbre and loudness, are more abstract than spatial position. Still, I claim that there exist concepts that can help us to establish a good match between the control space and the output space for DMI. This is the goal of the following section.

2.5 Connecting the control space with the output space

In this section we list several concepts which I deem important when forging a good connection between the control space and output space.

1. **Concept of effort and energy**
   With acoustic instruments you need to use some energy to get the wanted output and the amount of energy is usually related to the amount of sound you get, i.e. loudness. This is not necessarily the case for digital instruments since effortless actions can be mapped to sound with “unlimited” loudness. It has been suggested that users find the DMI responsiveness to be better if continuous input of energy is required for making continuous sound [23]. It has also been suggested that effort is closely related to expression [45].

2. **Concept of on and off**
   A concern with ubiquitous computing, e.g. computer systems that continuously interpret our actions, is whether an action is meant as a command or not. If we look at how

a performer plays an instrument, it is clear that it involves not only *sound-producing* actions, but also *sound-accompanying* actions, e.g., keeping track of the beat [28]. For this reason it seems important that the DMI design should keep some of the user's *action space* free. This gives the user some space to move in without interfering with the sound-producing actions.

3. **Concept of fault tolerance**
   If the input device is used for strict command-based events, it should be precise like a keyboard for text entry. Let us say that you want a *pattern recognition* system to recognize different command actions and that you can achieve a 90% recognition level. If this is intended to control important parameters you will soon get annoyed every time it does not recognize your actions correctly. This sort of imprecise control should only be used when accurate commands are not needed [21], i.e. such that small errors in the input or classification lead to only small and tolerable changes in the output.

4. **Concept of haptic feedback**
   Haptic feedback is often a physical property of acoustic instruments. This can be artificially integrated in digital controllers as *haptic technology* [58, 35]. However, it is not necessarily possible to implement such feedback in *virtual* musical instruments, i.e. instruments that are not based on physical controllers. An important question is what function the haptic feedback is intended to have. Is it just to give some feedback that an event is initiated or is it to express properties of the given state of the device?

5. **Bimanual input (Two handed input)**
   People use both hands in an asymmetric complementary way where the left and right hands have different tasks [21]. This is also the case when handling many traditional acoustic instruments. An awareness of this should be beneficial when designing DMI.

6. **Integral vs. separable dimensions**
   A computer mouse offers two integral dimensions while an *Etch-a-Sketch* toy offers two separable dimensions. While you have a good isolated control of each dimension with the Etch-a-Sketch, an isolated control of one the dimensions is more difficult with a mouse (see Figure 2.5). It has been shown that devices whose control space matches the perceptual structure of the task will enhance the performance for the user [26].

7. **Number of dimensions and degrees of freedom**
   When choosing an input device or a sensor, it will offer some number of control dimensions and an associated *degree of freedom*. These range, for example, from simple switch buttons that have one degree of freedom, on or off, to multidimensional continuous controllers. A match between the number of dimensions in the control and output space can be important [21].

8. **Absolute versus relative movement and position**
   A mouse measures relative movement while some motion capture systems, i.e. the electromagnetic tracker Polhemus, measure absolute position [25]. Again the DMI design will benefit from a choice of control space that fits the output space.

9. **Concept of responsiveness**
   An important property with musical instruments, which differentiates them from the field of HCI, is the role of *time* [60]. A great part of the musician's skill consists of properly *timing* musical events. In other words, high temporal precision can be an important feature for musical applications. Additionally, low latency is often seen as a prerequisite for achieving intimate control in musical interactive applications. The upper bounds for such control have been suggested to be 10 ms for latency and 1 ms for its variations, i.e. *jitter* [65]. We will return to these challenges in Chapters 3 and 4.

Figure 2.5: It is much easier to draw *integral* figures, e.g. diagonal lines, circles and bows, with a Wacom tablet (left) than with an Etch-a-Sketch (right). Yet, with the latter it is much easier to draw straight vertical and horizontal lines.

To clarify these concepts, let us briefly see how the *acoustic guitar* relates to them. First of all, the guitar offers a clear relationship between the energy spent when exciting the strings and the resulting loudness of the output (concept 1). It is obvious what *excites* the instrument and not, and the guitar offers many possible *sound-accompanying* actions. The strings can also be individually activated or dampened (concept 2). Furthermore, the guitar will never change the main behavior given similar control input. Any small variations in the given input will normally only give similar small changes in the output (concept 3). The guitar offers several layers of haptic feedback. The strings offer both resistance force when excited and vibration feedback after activation. The instrument body will also give feedback from its internal vibration (concept 4). The instrument offers a clear asymmetric complementary control space. Normally one hand controls the fretboard while the other is in charge of plucking and hitting the strings (concept 5). The guitar offers good separable control of each string. On the other hand, the fretboard can also be seen as combining the strings to one integral dimension, e.g. for barre chords (concept 6). The guitar offers further a clear perceptional image of the dimensions of the control space, normally 6 strings and a fretboard with about 20 frets, which has a direct mapping to the tonal output space (concept 7). All actions on the guitar affect also the guitar in a relative way, i.e. playing the guitar while hanging up-side down will not have any direct effect on the output. In other words, the guitar clearly defines and constrains the positional control space to its local coordinate system (concept 8). Finally, the guitar gives an immediate response to the user's actions, with no latency or jitter problems (concept 9).

Most acoustic instruments follow these concepts in similar ways because of physical constraints and the intrinsic behavior of the acoustic materials used. However, this is not the case

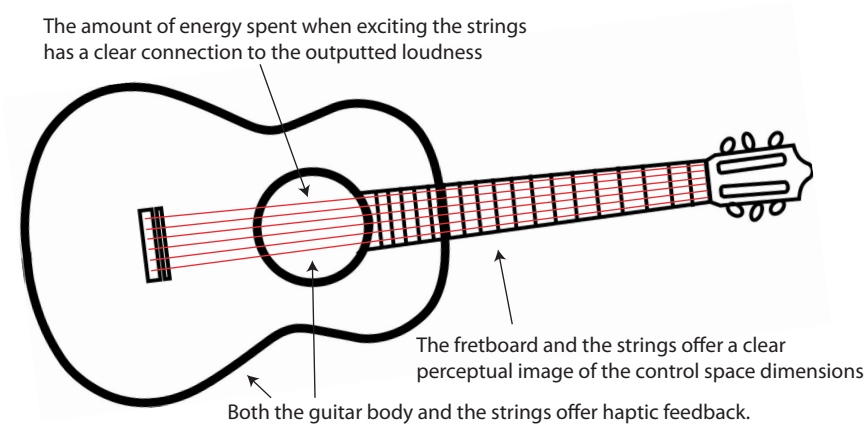


Figure 2.6: The acoustic guitar follows all of the listed concepts.

with digital instruments, since the action-sound mapping can be arbitrarily designed. In other words, these concepts must be explicitly incorporated in the design if we want the design to take advantage of our ecological knowledge of sound.

2.6 Discussion

It can be argued that many of the concepts listed above are merely ways of getting a device to become user-friendly, and that an artistic device is much more than to accomplish user-friendly aspects. This is an important point, and the usability should not be substituted for expressiveness and explorative qualities. In spite of this, the proposed concepts are, in my opinion, valuable guidelines to consider, since they support two important qualities of a DMI design, the *explorative quality* and *the communicative quality*.

The goal with usability in a wider sense is not only to make a task simpler, but to support spontaneity and momentum [22]. And I argue that not only will a device that is familiar in an ecological way be easier to explore, it can also increase the feeling of mastery and accomplishment. This can be important for the "flow feeling" of using a device, which is suggested to be important for joy [22]. In other words, the underlying idea is to design a DMI that supports user-friendly concepts which in the end are beneficial for the explorative quality of the instrument.

However, the concepts discussed are, in my opinion, not only beneficial for the performer, since the *intuitive instrument handling* can be shared with the audience. When I observe a musical performance, I am a curious spectator. If I cannot figure out the connection between the action and sound on stage, I easily become frustrated and bored by the performance. And it makes sense that we find it important to understand the connection between two of the most important modalities of a musical performance [3]. In particular, if we regard the performer's virtuosity as being an important factor enhancing the audience's experience, the audience's ability to comprehend the coupling between actions and sounds is helpful towards them perceiving the virtuosity on stage [57].

Several of the concepts listed above were actively used during the development of the *Dance Jockey system*. Since the system can be seen as a virtual instrument based on touchless motion and not on physical controllers, it was of great importance to build good couplings between action and sound. If the instrument is virtual, the whole comprehension of the instrument must come either from the sonic feedback or from the bodily experience of using the instrument. We found that the listed concepts made it easier to be conscious of how virtual instruments could be intuitively handled and perceived. We also found that the most interesting and successful mappings were made when these concepts were followed. Additionally, we wanted the spectators to benefit from these efforts, which was partly confirmed by the informal feedback we received after our performances. More details about the Dance Jockey system are presented in Paper IV.

Overall, it is difficult to reason that the discussed concepts of an instrument design can have any direct negative effect; however, they should not limit the designs. The instrument designer should indeed be free to incorporate counter-intuitive and surprising effects. The classic design quote "Know the rules well, so you can break them effectively", should be applicable in this respect.

2.7 Summary

In this chapter I have argued for some design considerations that I believe are applicable when designing digital musical instruments (DMI). I have introduced a simple conceptual model that I argue incorporates an important aspect of DMI designs. Based on this model, I have proposed a simple high-level design guide from which I think DMI designs can benefit. In effect, I suggest that the control space should somehow match the output space, and I discuss some concepts that a designer may take into consideration when attempting to connect these spaces.

Chapter 3

Motion Capture

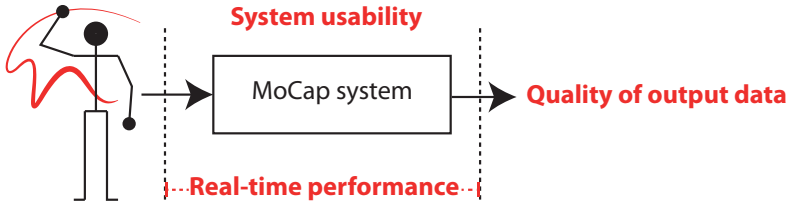


Figure 3.1: The task of the MoCap system is to capture motion. I have grouped what I see as the most important performance features of a MoCap system in three categories: *quality of output data*, *real-time performance* and *system usability*.

3.1 Introduction

Motion capture (MoCap) can be defined as the process of capturing motion and translating it to the digital domain. In this thesis we are especially interested in using the captured motion in real time for musical interaction. Since our goal is not to record the data per se, it might have been sensible to use the term *motion tracking* [64]. However, because of familiarity, I will in this thesis use the more commonly used term *motion capture* together with the established abbreviation *MoCap*.

The goal of the current chapter is not to give a comprehensive and thorough review of MoCap technologies and how they have been used in the field of DMI, but to present the essential challenges with MoCap and some additional details about the systems I have used in this thesis. I start by pointing out what I see as the main performance features of a MoCap system. Then I give a brief overview of the main technologies available before I finally present a summary and a discussion of the MoCap technology choices I have made for this thesis.

3.2 MoCap challenges

The main goal with a MoCap system is to *track* or *capture motion*. There are systems that only capture features of motion, for instance the distance between two objects or the acceleration of

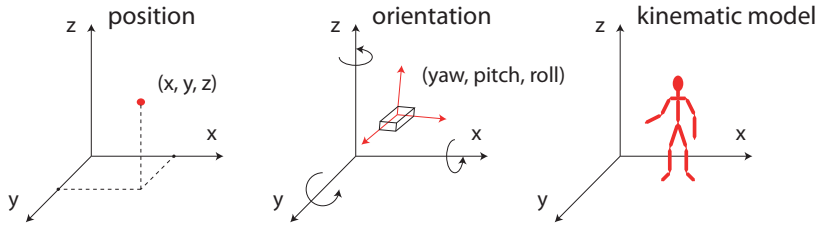


Figure 3.2: With a MoCap system we normally want to track the position of one or several objects. However, we may also be interested in tracking the orientation of objects, or a whole kinematic model, e.g. a skeleton model of a human being.

an object. These systems can give useful data in an affordable way. However, the goal with a MoCap system is normally to capture the position, and sometimes the orientation, of objects in space and time. It is also possible to track several *chained rigid objects* simultaneously. By grouping several rigid objects together and specifying their relative position and orientation, we can track kinematic models, e.g. a skeleton model of a human being, as illustrated in Figure 3.2.

Before we look into the details of how this can be done and the available technologies, let us start by considering the desired MoCap performance. The quality of a MoCap system can be evaluated in several ways. What may be an important feature for one application may be ignorable for other applications. In the following I will point out what I see as the most important performance features of a MoCap system. That is, how *spatially accurate* is the outputted data, how good is the *real-time performance*, and equally important, how *usable* is the MoCap system?

3.2.1 Data output quality - the spatial quality

The motion data we get from a MoCap system will normally have some deviation from the original physical motion that the data is based on. This can be seen as either noise or drift, where the former is seen as a random error, i.e. low precision, and the latter is more a continuous deviation which can compound over time. While some applications may need very accurate data, other applications can have other priorities. For instance, sub-millimeter resolution might not be the main priority when looking at body motions with an amplitude in meters. Low noise, robust and consistent data may be more important. As we will see later, there is no perfect MoCap system that fulfills every need, and it is therefore important to prioritize to be able to choose the most suitable MoCap systems for the required task [64].

Most MoCap systems work by sampling the sensor data, which are the basis for the data estimation, several times per second. As attested in the literature of biomechanics [68], and also supported by our work in Paper VII, the upper frequency content of human motion is normally limited to about 10–26 Hz. By following the *Nyquist–Shannon sampling theorem* a sampling frequency above 50–60 Hz should therefore capture the essential content of human motion [37]. However, higher sampling rates are positive for the resolution, since the samples can be regarded as noisy and inexact. Higher sampling rates can therefore give us increased resolution as long as this does not influence the system performance in other ways, e.g. reduced sensor

performance due to shorter exposure time during the sampling process. It is also reasonable to regard most MoCap systems as having so-called *white noise* properties, since they are based on sensor data which are regarded as having such noise distribution (see Section 4.5.1). Additionally, as the next chapter will show, if it is necessary to filter the MoCap data in real time, higher sampling rates lower the latency impact of the used filters. This brings us to the next important performance feature, the *tracking latency,* or the *real-time performance*.

3.2.2 The real-time performance

Since there are robust ways of accurately timing the sampling process, the original time stamp of the captured motion data is normally sufficiently exact. However, it takes time to process and transmit the required MoCap data to the end application [64], and the resulting *tracking latency* can be an unwanted feature for real-time musical interaction, as discussed in Section 2.5 under concept 9. An additional challenge is *jitter*, i.e. the variation of the latency, which is an important feature if high temporal precision is needed. In other words, the problem with distortion in the time domain, is normally not when the data was captured, but when the data is received by the end application, as illustrated in Figure 3.3. *Buffering* can be used to minimize the jitter problem, but this will increase the overall tracking latency [48]. Notice that such distortion of the time domain has a negative effect on the *spatiotemporal* accuracy.

A contributing factor for the above problem is that commercially available computers and network systems do not support streaming of real-time data with minimal latency and jitter performance. Even if the MoCap system could support the delivery of data with low jitter and latency, it would still be a problem to transmit the data with standard computer platforms like WIFI, Bluetooth, Ethernet, etc. However, the new *Ethernet AVB* protocol may solve some of these issues [48]. Another related problem is so-called *frame drops*, i.e. that the MoCap or network system is not capable of sending every sampled time frame. Not only is this critical since we can miss out on important actions, it is also problematic when *differentiating* the motion data, i.e. calculating the derivative. Missing samples can result in value leaps in the

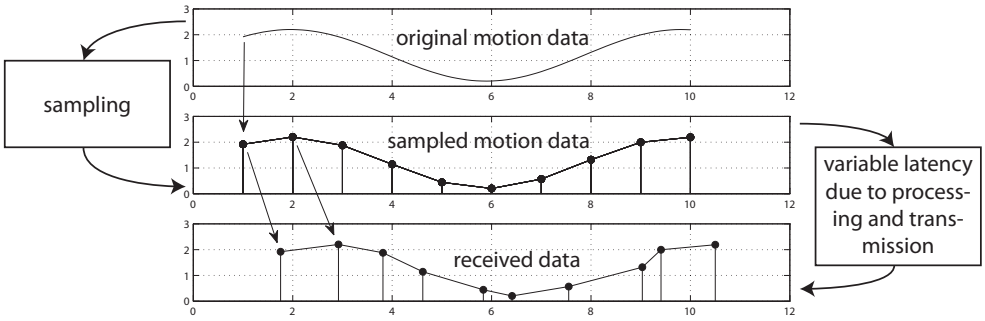


Figure 3.3: Illustration of the time domain challenge of using MoCap data for real-time applications. Though the original data is correctly sampled in the time domain, it takes time to *process* and *transmit* the data to an end application. A variation in the latency results in *jitter*, which can be seen as a distortion in the time domain, as shown in the lower curve (notice the distorted waveform).

received data streams, which will result in large differentiation errors if not properly taken care of.

In other words, though a system can offer very accurate motion data, it will not necessarily be suitable to incorporate in a DMI design if it has poor real-time performance. In similar ways, a high-end MoCap system may have limited use outside the lab due to other practical usability concerns.

3.2.3 Usability and the "out of lab" performance

The final quality of a MoCap system is not determined by the technical performance which can be achieved in a lab, but how it works in practice for the end application. It is therefore important to consider how the system performs for the intended use. In the following, I list what I see as the most important usability features of a MoCap system.

- **Environmental "robustness"**
  While a system may work perfectly in the lab, it may perform poorly in a different environment. Thus, it is important that the MoCap system performs well in the intended environment. In other words, the sensors used must be satisfactorily immune to the given environmental interference, e.g. stage lights, electromagnetic interference, temperature shift.

- **Tracking area.**
  The system needs to deliver the wanted performance for the whole of the required tracking area. While some systems only work for very small areas due to limitations of the sensors used, e.g. optical systems, *inertial* systems can work in an unlimited area.

- **Obtrusiveness**
  It is important that the system used is not too obtrusive for the performer. A MoCap suit can affect the performer's ability to move if it is cumbersome to wear, e.g. a heavy suit involving multiple cables. A large and visually distracting system can also interfere aesthetically with the performance.

- **Portability and setup time**
  While some systems can fit in a pocket, e.g. the Nintendo Wii Remote, other systems may have greater transportation needs. The complexity of the system affects also the mounting and unmounting time required. These features determine the practical sides of touring and traveling with the system, i.e. when used for multiple locations over short time periods.

- **Number of tracked objects or subjects**
  Due to system limitations, e.g. processing power or network bandwidth, the tracking performance may be heavily influenced by the number of tracked objects or subjects. It is therefore important to use a system that supports robust tracking of the desired number of objects or subjects.

- **Reliability - robustness and stability**
  Finally, it is important to consider the overall reliability of the hardware and software.

Software “bugs” and badly engineered hardware can make the system frustrating to work with. The quality of the hardware determines not only technical features like battery lifetime but also how solid and robust the system is in the long run, i.e. the life expectancy.

Let us now go through some of the available MoCap technologies, and how they relate to the performance features discussed above.

3.3 Available MoCap technologies

There are several available MoCap systems on the market today, all with their different strengths and weaknesses and intended use. There are mass-produced systems that come from the computer game industry with an affordable price tag. At the other end, there are specialized high performance systems with very high price tags, which limits their use to industry and research institutes. However, all MoCap systems are based on sensors. The data from these sensors is analyzed in different ways to be able to make a good estimation of the spatial properties of the tracked object. The capture quality is therefore dependent on the quality of the sensor systems and analysis methods used. In the following section, I will list the main available technologies.

3.3.1 Optical systems

The earliest form of motion capturing was simply using our own vision. The invention of photography and cinematography made it possible to perform more objective and precise tracking of motion. Placing markers on the tracked objects allowed for somewhat precise manual estimation of properties like speed and acceleration [68]. The adaptation of the digital camera made it possible to automate these processes on digital computers. Essentially, optical systems rely on optical measurements of reflected or emitted light. In other words, these systems consist of two components: light sources and optical sensors. We can divide them into two different subcategories, *marker-based* and *marker-less* systems.

Optical *marker-based* systems

The optical marker-based system is today one of the most accurate MoCap systems available and can achieve sub-millimeter resolution. It works by using digital cameras in combination with markers that are placed on the tracked object(s). By utilizing infrared cameras and light sources, it is possible to operate within a light spectrum that does not interfere with our own vision. This makes the system also somewhat less prone to light pollution. It is further possible to use either active or passive markers. Active markers emit light themselves, while the latter work by using a light source on the cameras in combination with reflective markers (see Figure 3.4).

Using one camera, it is possible to measure how one or several markers move in the 2D view frame of the camera. If the size of the measured marker is known, it can be used to roughly estimate its distance from the camera. However, more accurate and precise three-dimensional positions can be estimated by *triangulation* if two or more cameras can see the same marker. Additionally, a rigid object’s *orientation* can be estimated if three or more markers are placed

Figure 3.4: The two main MoCap systems that have been used during the work of this thesis are the OptiTrack V100:R2 (left) and the Xsens MVN system (right). Notice the IR LEDs on the OptiTrack camera which are used as the light source to light up reflective markers. The strap-on suit, on the far right, is the Xsens suit we have used for the Dance Jockey project.

on the object. And, if the placement of the markers on the rigid bodies is done in a unique way, it can be used to identify the objects. In this way, a system can track and identify several rigid objects in the capture area, and can be used to track a complete kinematic model, e.g. a human body.

Multi-camera MoCap systems need to be calibrated before use. The calibration process determines the position and orientation of the cameras and is the basis of how the camera estimates the position of the markers. It is therefore necessary to perform a new calibration if the camera setup is accidentally distorted after the calibration process, i.e. if the position or orientation of the cameras is accidentally changed.

The main benefit of optical marker-based systems is the possibility of very accurate positional tracking and fairly high sampling frequencies. The resolution of the camera sensors used and the proximity to the marker determine the possible tracking resolution. These systems can also track multiple markers and objects simultaneously, as long as the markers are visible to the cameras. *Optical occlusion*, i.e. when markers are temporarily out of sight of one or several cameras, can be seen as the system's main drawback which can cause *frame drops*, *marker swap* and *occlusion noise*. The latter noise occurs when a marker's position is estimated with different sets of cameras during the tracking session due to optical occlusion. This will result in slightly different position estimates and hence noise (see Paper III). While these occlusion problems can be fixed in post-processing software, real-time data will suffer from inconsistent and noisy data. It is therefore important to have a good distribution of the cameras in the tracking area to minimize marker occlusion problems. This again demands multiple cameras, long wires, heavy tripods and time-consuming preparations. And, though they normally work in the infrared spectrum, they are still sensitive to light pollution since many light sources contain infrared light.

Optical *marker-less* systems (Computer vision)

Computer vision-based systems are essentially marker-less optical systems that rely on digital image processing techniques to recognize objects, position, motion, activity, features and more. While they do not offer the same accurate positional tracking ability as marker-based systems, they avoid the use of obtrusive and cumbersome markers. Computer vision-based systems are, similar to optical marker-based systems, prone to optical occlusion and pollution. In spite of

this, it is a promising technology which can potentially be very versatile and affordable. In its simplest form, a system can consist of a web camera and some analysis software running on the attached computer. However, these systems can also be multi-camera based. *Stereo vision* is a much used approach which is based on two cameras for improved 3D estimation, similarly to our own stereo vision capabilities *(stereopsis*), e.g. *leap motion* [63]. There are also several systems that have more sophisticated built-in sensors to improve the estimation of different features, e.g. Microsoft Kinect's *depth sensor* [71] and the new Xbox One *time-of-flight sensor* [20].

3.3.2 Inertial systems

Unlike optical systems that rely on external observation, inertial systems estimate motion without the need for external references. For some applications this can be very practical since they are not dependent on external sensors or systems, i.e. they are *self-contained*. Inertial sensors are based on *inertia*, i.e. the resistance of any physical object to change in its current motion. One of the most popular inertial sensors is the accelerometer. While it is possible to use an accelerometer alone to do some basic motion analysis, it is not possible to perform robust spatial estimation since the orientation is unknown. However, by combining an accelerometer with a gyroscope, it is possible to calculate the position, orientation, and velocity of the attached object via *dead reckoning*[1] [64]. To combine several sensors in this way is often referred to as *sensor fusion*.

Kalman filters are often used in these applications to minimize positional and orientational estimation errors [52]. Basically, the position and orientation are estimated by integration of angular velocity measurements from gyroscopes and double integration of accelerometer data. Given that these sensors give noisy results, it is necessary to use some kind of noise filter to improve the estimations. Kalman filters are so-called recursive filters that produce statistically more optimal estimates by having knowledge of the underlying system. Nevertheless, the position estimation of such systems drifts several meters in a short amount of time due to imperfect sensors [64].

While inertial systems earlier had only limited use due to large and expensive sensors, the adoption of *microelectromechanical systems* (MEMS) has made it possible to make very compact inertial sensors [64]. These MEMS sensors, due to their affordable price, have become standard in many consumer devices like mobile phones and computer game controllers. Such inertial systems do not offer the same accurate tracking quality as optical marker-based systems, and they are especially prone to positional drift. On the other hand, they offer a self-contained MoCap technology without occlusion problems and with a theoretically infinite tracking area. These sensors can also be sampled at high sampling rates [64]. The reduced accuracy (i.e. drift) of MEMS sensors can be compensated somewhat by using compact reference-providing sensors like magnetometers and GPS sensors. However, these resulting systems are no longer strictly inertial.

It is possible to use several of these sensor systems in parallel to track the motion of a complete kinematic model, such as a human body. The tracking quality of such systems can

[1] *Dead reckoning* is the process of calculating an object's current position by using a previously determined position and advancing that position based upon estimated speeds over elapsed time.

be improved by fitting the sampled sensor data to a biomechanical model of the tracked subject [43]. Today there are several commercially available MoCap suits that are based on inertial sensors. I have used one of these systems, the *Xsens MVN suit*, shown in Figure 3.4, for the *Dance Jockey project*. See Paper II for more details about this MoCap suit.

3.3.3 Other available MoCap systems

There are several other sensors that can be used to capture positional and orientational motion properties. In the following section, I will list the main available technologies. These technologies have not been used during the work of this thesis. Therefore only a brief overview is given. See [64] for a more detailed overview of these systems.

- **Mechanical systems** are based on sensors that sense mechanical motion and forces directly, e.g. potentiometers and bend sensors. This can result in affordable and effective systems for some applications. However, as one might expect, it can easily lead to quite obtrusive systems when used for complete tracking of the full human body. Nevertheless, they can offer very precise and intimate control since the analysis of the sensors used is normally straightforward.

- **Magnetic systems** utilize sensors that can *estimate spatial properties* based on either Earth's magnetic field or an active coil that emits a strong magnetic reference field. Given Earth's weak magnetic strength, the former systems are very sensitive to magnetic disturbance [64]. With an active coil it is possible to achieve very good occlusion-free and complete six-dimensional tracking, i.e. the position and the orientation of several objects in a compact system. However, active coil systems are also prone to electromagnetic interference and their tracking range is very limited because of the cubic decrease of magnetic field with the distance to the source [64].

- **Acoustic and radio frequency (RF) systems** work by evaluating the attributes of a target by interpreting the echoes from radio or sound waves. In this way, they can measure the distance to one or several objects. The *wavelength* of the transmitted wave determines the achievable *resolution*. Both systems therefore have somewhat restricted use, since they are limited by the physics of the waves used. Acoustic systems are mainly based on ultrasound sensors, given the short wavelengths. RF positioning systems are becoming more viable as higher frequency RF devices (i.e. shorter wavelengths) allow greater precision than older technologies. However, both types are susceptible to interference in the environment and none of these systems can compete with the sub-millimeter accuracy of optical or magnetic systems. Nevertheless, they have some attributes that can be beneficial for some applications; for example, RF systems can work in a large capture area [64].

- **Hybrid systems** are important to mention when giving an overview of available MoCap technologies. The essence is to use several different complementary sensors that can together offer the required tracking resolution and performance that best facilitates the given application. Several commercially available systems are based on this strategy, e.g. the Wii Remote (accelerometer combined with an optical system - Sensor Bar, and

the expansion of gyroscope functionality with the Wii Motion plus), PlayStation Move (accelerometer, gyroscope, magnetometer and optical marker-based tracking through a camera).

3.4 Discussion

As we have seen, there are several MoCap technologies available, and though no *silver bullet* exists [64], i.e. no one technology that satisfies every need, they offer together a usable set of tools for tracking motion with their different strengths and weaknesses. The optimal technology is therefore dependent on the priorities of the targeted application. During the work of this thesis, I have mainly used the optical marker-based system known as OptiTrack V100:R2, and the already mentioned inertial sensor based suit, the Xsens MVN system. Both systems are shown in Figure 3.4.

I soon became aware of the Xsens MVN suit's potential as a portable, practical and robust all-in-one system. The usability of the Xsens system had several benefits compared with the more accurate optical marker-based systems. The portability and the fast setup time of the Xsens MVN system made it possible to stage performances in different locations without the need for much logistics and tedious preparation. However, most importantly, since the Xsens system does not suffer from *optical occlusion*, the real-time MoCap data was much more consistent and robust, as shown in Paper III. Though the output data is less accurate than what the optical system typically offers, the robust real-time performance was more important during the *Dance Jockey project*.

I have also seen the advantages of optical marker-based systems when it comes to accurate data and the simplicity of doing several subsequent recordings with a limited amount of markers. While it easily takes more than 10 minutes to put on and calibrate the Xsens MVN suit, placing a limited amount of reflective markers on a subject can be done in seconds. It is also easier to get volunteers for an experiment when it only involves wearing a few markers as opposed to putting on a cumbersome suit. An additional benefit of these systems is that they output raw non-filtered data (see Paper III). The Xsens system is based on several layers of processing steps to make the best possible positional estimations [43]. However, these processing steps can also distort the estimated positional data. It is difficult to take these distortions into account in an experiment, since the proprietary processing steps are normally hidden from the user. In my opinion, these features make the optical marker-based systems more suitable for quantitative experiments, for instance, measuring the maximum frequency of free hand motion, as presented in Paper VII.

On the other hand, since the Xsens MVN system is based on accelerometers, it can output acceleration data directly based on measurements of these sensors. With optical marker-based systems, acceleration data need to be calculated from the positional raw data through differentiation. This process both adds latency and can increase the noise problems in the data, which again demands extra noise-smoothing. The challenges with noise-smoothing and differentiation of MoCap data are the subject of the next chapter.

3.5 Summary

In this chapter I have given a brief overview of the challenges with motion capture technologies. I have grouped the performance of MoCap systems into three categories, *quality of the data*, *real-time performance* and *system usability.* I have further presented some of the main technologies available, and detailed some of the characteristics of the different systems. Finally, I discussed and argued for the MoCap technology choices that have been made in this thesis.

Chapter 4

Filtering MoCap data

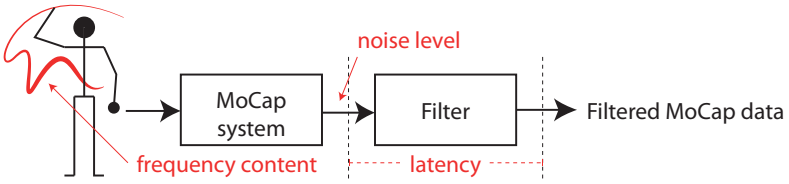


Figure 4.1: Important terms when wanting to design suitable MoCap filters for real-time applications.

In this chapter I present some of the basic details of digital filter theory and summarize my work of studying best practices for filtering MoCap data for real-time applications. However, some of the content of this chapter should be equally valuable for those that want to filter MoCap data for post-processing purposes, i.e. when the real-time properties are not important.

4.1 Introduction

Digital signal processing (DSP) can be defined as the mathematical processing of a signal, with the intention to *improve* or *modify* the signal in some way. The most common processing approach in the time domain is through a method called *filtering*. As we have seen in the previous chapter, many of the utilized MoCap and sensor technologies are known to possess noise properties that may be problematic (see Paper III) [68]. It is therefore often necessary to apply *noise smoothing filters* to alter these noise problems. However, *noise-smoothing* is not the only interesting utilization of digital filters. We can also perform *feature extraction,* i.e. transforming the MoCap data in some way that makes it more interesting and useful. In this section I discuss appropriate methods for *noise smoothing* and *differentiating* MoCap data. Differentiators can be used to extract *velocity* and *acceleration data* from positional MoCap data, which, together with *position*, were experienced to be some of the most useful motion features for our target application during the Dance Jockey project.

As already pointed out, low latency can be an important property for achieving intimate control in musical applications [65]. And, as one might expect, there will always be a corresponding delay penalty when employing a *digital filter*. There exist several established methods for designing digital filters for noise smoothing and differentiation [37]. However, none of them

are suitable for designing filters with minimal delay properties, as described in Paper V. Consequently, it was necessary to find other ways of designing such filters. An alternative design method was therefore developed during the work of this thesis. With this method, it is possible to design more optimal low-delay filters than the currently available design methods can produce. The proposed design method, including a range of different low-delay filter designs, is a significant part of the contribution of this thesis. An important focus of this section is therefore concerned with comparing the delay performance of different filter design methods.

In the following section an introduction to digital filters is given. I then continue by presenting some filter analysis methods and filter design methods in Sections 4.3 and 4.4. In Section 4.5 I discuss methods for determining reasonable cutoff frequencies when filtering MoCap data. Then in Section 4.6 I give a comparison of the delay performance between different filter design methods, and in Section 4.7 I give some additional comparison details. Finally, in Section 4.8, I give a discussion and summary of this chapter.

4.2 Digital filters

A common goal when applying filters is to smooth or restore data that have been distorted with noise. There exist several methods, and they can roughly be divided into two categories: *curve fitting techniques* and *digital filters* designed in the frequency domain. Curve fitting can be intuitively explained as trying to graphically fit a smooth curve to noisy data. The most common methods are *polynomial fit* and *spline methods* [68]. However, curve fitting noisy MoCap data is known not to be optimal since human motion does not necessarily follow polynomial curves [42, p. 235]. Digital filters that are designed and evaluated in the frequency domain are seen as the most general method for noise smoothing and are the tools I have used in this thesis. This should also be the most sensible choice since we need filters with *causal* behavior and good real-time properties. Causal behavior indicates that the filter output depends only on past and present inputs, i.e. a mandatory property for real-time applications.

When discussing digital systems, it is common to limit the discussion to so-called *linear time-invariant* (LTI) systems, which demand that the given system needs to be *linear* and *time invariance,* i.e. that the time does not affect the output given the same input. All filters discussed in this thesis are LTI systems [37].

4.2.1 A digital signal

Most MoCap systems offer motion data in a digital format. This means that the output data is a sequence of discrete values that represent a continuous physical signal. The sampling frequency f_s, given in Hertz (Hz), indicates how many times per second the signal is sampled. It is important to band-limit the signal to *half* of the used sampling frequency before converting it into a digital signal. A digital signal with a sampling rate of 100 Hz cannot contain frequencies higher than 50 Hz, i.e. half the used sampling frequency (see the *Nyquist sampling theorem* [37]). In other words, when showing the frequency content of a digital signal, it is normal to only show this possible range, i.e. from 0 to 50 Hz, as shown in Figure 4.2. Furthermore, since digital systems can be used with different sampling frequencies, the relation to time is not fixed. Digital systems are therefore often specified in *normalized frequency*, denoted as ω. To

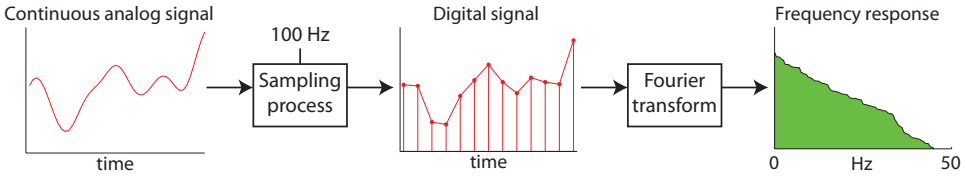


Figure 4.2: Illustration of how a continuous analog signal is converted into a digital signal. With the Fourier transform, it is possible to show the frequency content of a digital signal.

convert normalized frequency to Hertz, multiply by half of the used sampling frequency, i.e. $f = \omega \cdot f_s/2$.

A fundamental tool in DSP is the *Fourier transform,* which makes it possible to express the *frequency content* of a digital time domain signal. We will primarily use the frequency domain when designing filters, as the following section explains.

4.2.2 Noise smoothing with *low-pass filters*

Formally, the goal of a noise filter is to extract the desired signal from some noisy data. Typically, this is done by designing a filter with the purpose of removing the noise component while leaving the desired signal unchanged. This is the classical purpose of low-pass filters. These filters pass low-frequency signals while suppressing or attenuating high-frequency signals, as illustrated in Figure 4.3. This strategy works for MoCap data since human motion mainly consists of low frequency signals [68]. The *passband* refers to those frequencies that are passed, i.e. wanted, while the *stopband* refers to the frequencies we want to filter out. To not distort the passband, it is necessary to have a constant gain, i.e. a flat magnitude response, in the passband. In order to maximize the noise attenuation, it is necessary to have the lowest possible gain in the stopband. *Moving average* is probably the most simple and intuitive realization of a low-pass filter. Moving average is frequently used because it is intuitive and simple to implement. While these filters have low-pass filter properties, the *magnitude response* in the frequency domain is solely specified by the order, i.e. the length, of the filter, as illustrated in Figure 4.4. As we will see, in many cases there are more optimal filter design solutions. Notice that the *magnitude response* specifies how the filter amplifies or attenuates a signal in the frequency domain.

A common way to design more sophisticated digital filters is to optimize how they perform

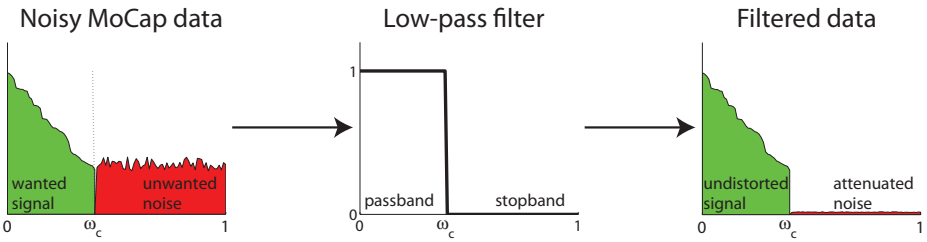


Figure 4.3: Illustration of the purpose of a low-pass filter in the frequency domain. By using a suitable *cutoff frequency* ω_c, it is possible with a low-pass filter to suppress the unwanted high-frequency noise of the input signal while preserving, i.e. not distorting, the wanted content.

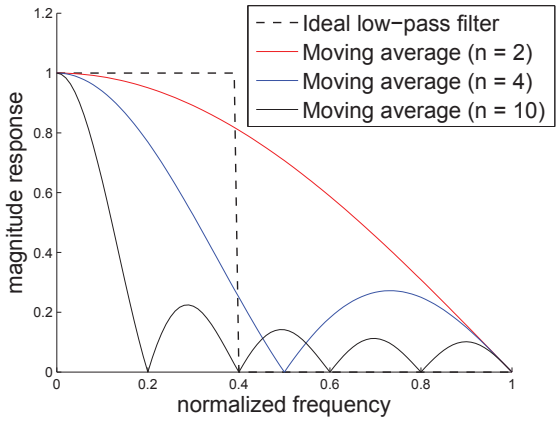


Figure 4.4: The magnitude response properties of three moving average filters of orders 1, 3 and 9. Moving average filters have low-pass filter properties but deviate from the ideal low-pass filter response, in this example with a cutoff frequency of $\omega_c = 0.4$.

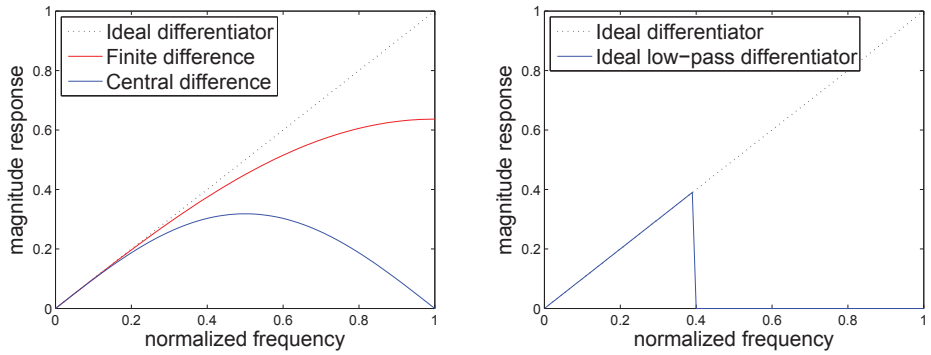


Figure 4.5: The magnitude response of an *ideal differentiator* of *degree 1* together with the magnitude response of the *finite difference* and the *central difference* implementations (left) and an ideal *low-pass differentiator* with a cutoff frequency of 0.4 (right).

in the frequency domain. This consists of determining the localization of the passband and stopband in the frequency domain and designing an appropriate filter based on these properties. Before we continue with presenting how such filters can be designed, let us first consider a related filter design challenge.

4.2.3 Low-pass differentiators

Differentiators are a filter type that can be used to extract velocity and acceleration data from position data. This is a much-used operator since most of the available MoCap systems offer only spatial, i.e. positional and orientational, motion estimations. If a property like velocity or acceleration is wanted, it is necessary to use differentiators to compute the derivative of the spatial data. The frequency response of an ideal differentiator is a linear line in the frequency

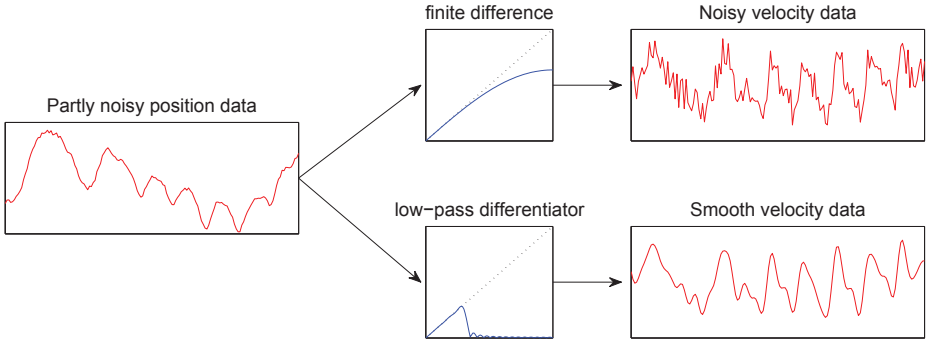


Figure 4.6: By using a low-pass differentiator, we avoid the undesirable amplification of noise in the higher-frequency band.

domain, as illustrated in Figure 4.5 ($G(\omega) = \omega$). The interpretation of this is that lower frequencies have a lower rate of change, while higher frequencies have a higher rate. (Higher velocity and acceleration are needed to get an object to oscillate with a higher frequency while preserving the amplitude.) The simplest digital implementation of a differentiator is the finite difference equation, given by

$$y(t) = \frac{(x(t) - x(t - \Delta))}{\Delta}, \tag{4.1}$$

where Δ denotes the time distance between samples. While this implementation follows the ideal response for low frequency, it deviates for higher frequencies ($\omega > 0.4$), as shown in Figure 4.5. The central difference equation

$$y(t) = \frac{(x(t + \Delta) - x(t - \Delta))}{2\Delta} \tag{4.2}$$

deviates even more (from $\omega =\sim 0.2$) and goes to 0 for $\omega = 1$. In other words, these implementations do not follow the ideal differentiator response for the whole frequency band. However, this can actually be a wanted feature. When differentiating MoCap data, it is normal to experience an increase of noise in the differentiated data. This is due to the fact that differentiation resembles a *high-pass* filter, as can be seen by the ideal differentiator curve in Figure 4.5. That is, the low-frequency motion data in the passband are attenuated, while the noise in the higher frequencies are amplified. As a result, we end up with having more noise in the differentiated data, which increases the need for noise filtering [68, 15]. This is why it is reasonable to use *low-pass differentiators* since they only follow the ideal differentiator curve in the passband and avoid the undesirable amplification of noise in the higher-frequency band, as shown in Figure 4.6. They will also provide more optimal total filter solutions than using a low-pass filter in cascade with a differentiator operator, e.g. finite difference, as we have shown in Paper VI. It is also possible to design a low-pass differentiator *of degree 2* (or higher) which can be used to compute the double derivative directly instead of using two differentiators in cascade.

4.2.4 Filter objectives

We have now described our two main filters, *low-pass filters* and *low-pass differentiators*, which will be the focus of this chapter. They have several features in common. Both filters have low-pass filter characteristics, but with different wanted passband behavior. Their two main filter objectives are as follows:

- *To minimize the passband distortion*. That is, we do not want the filter to alter the desired output, but to follow the wanted response in the passband.
- *To maximize noise attenuation*. That is, to reduce the amount of noise as much as possible.

There are established filter design methods that satisfy these two objectives [67]. However, as I already have mentioned, in this thesis I am especially interested in the following additional objective:

- *To minimize the filter delay*. That is, to minimize the time it takes for the signal to pass the filter.

Let us now go through some filter theories, which will make it possible to analyze and design more sophisticated filters than the ones presented above.

4.3 Filter analysis

The goal of the following section is to go through some of the filter theories, which is necessary in order to be able to compare digital filters. I will explain the main difference between the so-called FIR and IIR filters and how they can be designed. Let us start by explaining the impulse response of a filter, which can be an intuitive approach to understanding the workings of a digital filter.

4.3.1 The impulse response

There exist two main digital filter types: *finite impulse response* (FIR) filters and *infinite impulse response* (IIR) filters. Before giving a formal description of these filters, notice that the impulse response is specified to be the key difference between these two filters. The *impulse response* of a filter is the given output when presented with a brief input signal called an impulse. This response gives us a time domain view of how the filter works. For an FIR filter, the relation is straightforward since the impulse response corresponds directly to the filter coefficients. The *moving average* FIR filter, as the name suggests, works by setting the output $y[n]$ to the *average* of a subset of samples, or a window, of the input signal $x[n]$. Every new output $y[n+1]$ is calculated by moving the window one step, i.e. one sample, further. The longer the filter, the more samples are used in this average estimation (which also provides more noise attenuation as shown in Figure 4.4). While the moving average can have wanted features for some conditions, it is possible to use more sophisticated weighting coefficients that result in a more ideal magnitude response, i.e. with a specified passband and stopband, as shown by the FIR design in Figure 4.7.

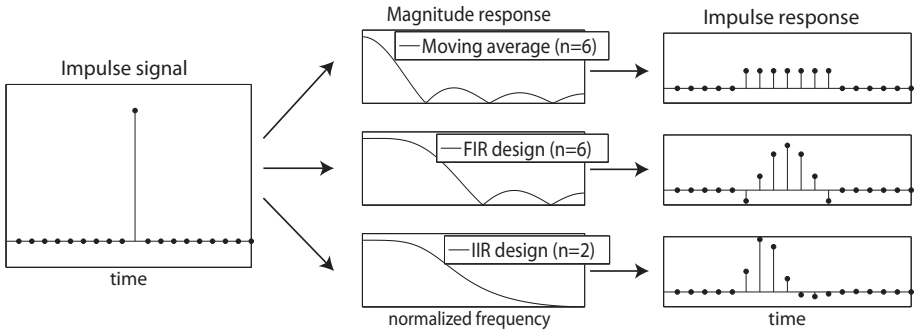


Figure 4.7: Comparison of the impulse responses of three different low-pass filters. The magnitude responses of these filters are shown in the middle.

However, we are not only concerned with the magnitude response when designing a filter. The *phase response* can be equally important and normally is a *linear* phase response in the passband wanted. A linear phase will not distort the phase of a signal. That is, a linear phase will ensure a *symmetric impulse response* [54]. It is easy to ensure a linear phase for a FIR filter since it simply involves checking if the filter coefficients are symmetric. Notice that the IIR design in Figure 4.7 does not have a symmetric impulse response.

In Figure 4.7, we can recognize how the filters delay a signal by looking at the impulse response. We can loosely say that the delay corresponds to how long the impulse response takes to rise to the maximum amplitude. The delay of a symmetric FIR filter has a simple relationship with the filter order n and is given by $n/2$ samples. This corresponds well with the impulse responses of Figure 4.7 and is especially visible for the FIR design in the middle. Furthermore, it can be seen that though the IIR filter has a similar magnitude response as that of the FIR filter, it delays the impulse with only about one sample compared with three samples for the FIR filter. In essence, IIR filters offer an effective way of achieving a long *impulse response*, without having to use long FIR filters. Therefore, if the goal is to minimize the filter delay, the use of IIR filters seems reasonable since they can have a dramatically lower order than symmetric FIR filters with similar performance [37]. Our results in Paper V and Section 4.6.2 support this claim as well. However, as you might suspect, designing IIR filters with linear phase, i.e. symmetric impulse response, can be more challenging.

Notice that the impulse response determines and specifies how the filter works. It is therefore possible to transform an IIR filter into a FIR filter by using the impulse response of the IIR filter directly as the FIR filter coefficients. The more coefficients we use, the closer we get to replicate the exact frequency response of the given IIR filter. FIR filters that are not symmetric are known as *asymmetric* FIR filters. Let us now continue with presenting some filter theories, which will enable us to perform better filter comparisons.

4.3.2 FIR and IIR filters

The output of a FIR filter is a weighted sum of the current and finite number of previous values of the input. The operation can be described by the following equation, which defines the output

sequence $y[n]$ in terms of its input sequence $x[n]$:

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots + b_Nx[n-N] = \sum_{k=0}^{N} b_kx[n-k]. \tag{4.3}$$

Here, b_k are the *filter coefficients* and N gives the *filter order.*

IIR filters, as the name suggests, have an *infinite impulse response* that is the result of their recursive structure. While a FIR filter only bases its output on the input signal $x[n]$, an IIR filter bases its output on former output values $y[n]$ as well:

$$a_0 \cdot y[n] = \sum_{k=0}^{N} b_kx[n-k] - \sum_{k=1}^{N} a_ky[n-k]. \tag{4.4}$$

The goal of a filter design is to find a set of filter coefficients a and b that corresponds best to our filter needs. The following question then arises: how are these coefficients related to filter performance?

4.3.3 The transfer function

The most common way of analyzing a digital filter is through a mathematical analysis of the *transfer function [37].* Without going into the details, the above digital filter Equation (4.4) can be expressed through the *Z-transform* [37] as the following *transfer function:*

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1z^{-1} + \cdots + b_Nz^{-N}}{a_0 + a_1z^{-1} + \cdots + a_Nz^{-N}}. \tag{4.5}$$

For FIR filters, the coefficients a_k will be 0 for $k > 1$. Notice that a_0 is a gain coefficient which is normally set to 1, as shown in Equation (4.4). While Equations (4.3) and (4.4) explain how the filter works in the time domain, Equation (4.5) expresses how the filter works in the frequency domain. Through this transfer function $H(z)$, which is often rewritten as $H(e^{j\pi\omega})$, where ω denotes the *normalized frequency*, we have a powerful tool for filter analysis, since it is possible to express both the *magnitude* response and the *phase* response. The magnitude and the phase response are, respectively, the absolute value and the complex part of $H(e^{j\pi\omega})$. That is, the absolute value of the transfer function $H(e^{j\pi\omega})$, often written as $H(j\omega)$, gives the magnitude gain for the normalized frequency ω.

$$G(\omega) = |H(j\omega)| \tag{4.6}$$

Meanwhile, the complex part gives the change in phase.

$$\theta(\omega) = \arg(H(j\omega)) \tag{4.7}$$

In other words, by inserting the different filter coefficients in the transfer function in Equation (4.5), we can calculate the *frequency response* and *phase response*. Instead of referring to the phase delay, I will use the term *group delay*, which indicates how many *samples* certain frequencies are delayed by the filter. The group delay is found by computing the negative derivative of the phase shift with respect to normalized frequency ω (i.e. the more it shifts, the

more it delays, and if the shift is linear, the group delay will be constant).

$$\tau_g(\omega) = -\frac{d\phi(\omega)}{d\omega} \tag{4.8}$$

While the above results are somewhat mathematical, it is possible to show a more intuitive relationship between the magnitude response and the transfer function by considering the roots of the nominator and denominator polynomial in *the z-plane.*

4.3.4 The z-plane

A *z-plane* plot gives us a visualization of how the transfer function affects the magnitude response and how the choice of different coefficients gives different results. The roots of the numerator and the denominator of the transfer function, known respectively as the *poles* and *zeros*, can be plotted in the z-plane, as shown in Figure 4.8. The interpretation of how the poles and zeros affect the magnitude in the frequency domain can be found by regarding the *unit circle* in the z-plane. The resulting frequency response of the filter is related to how these poles and zeros influence the unit circle, as illustrated in Figure 4.8. The zeros are responsible for attenuating the magnitude response, while the poles are responsible for the amplification. The closer the zeros and poles are to the unit circle, the greater the effects they have on the final magnitude response. Notice that poles and zeros are symmetric about the real axis, which is a requirement for a real filter.

Filter design is essentially about choosing an optimal and balanced placement of zeros and poles that satisfies the wanted filter response the most. The number of poles and zeros correspond to the *filter order*. Higher orders, i.e. more poles and zeros, give us more potential to shape the wanted magnitude response. While IIR filters can move their poles around in the z-plane, as shown in Figure 4.9, FIR filters have all their poles fixed to the origin of the z-plane. This makes IIR filters more customizable for the same filter order, which is the essential

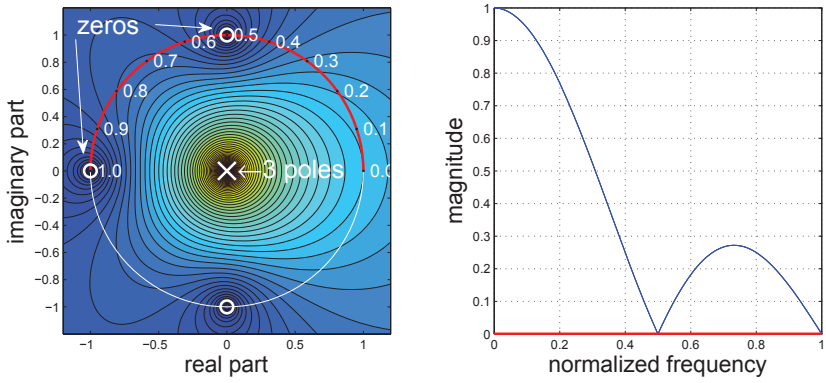


Figure 4.8: A contour plot of the z-plane plot (left) and the magnitude response (right) of a *moving average* filter of order 3 ($\mathbf{b} = [1\,1\,1\,1]/4$) and how they are related. Notice how the placement of the zeros , i.e. $z = -1$ and $z = 0 \pm i$, on the *unit circle* in the z-plane affects the magnitude response. All three poles have a static position in the origin since this is a FIR filter.

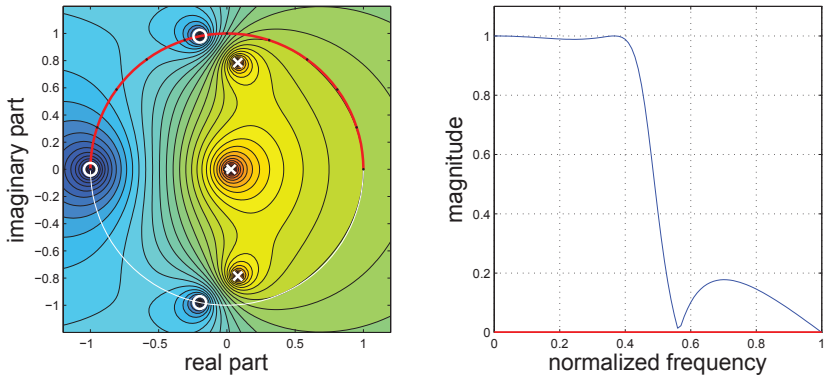


Figure 4.9: A contour plot of the effect of poles and zeros in the z-plane (left) and the corresponding magnitude response (right) of an *elliptic* filter design of order 3. Notice how the poles are spread out, which results in a very flat passband response, i.e. with a low passband distortion.

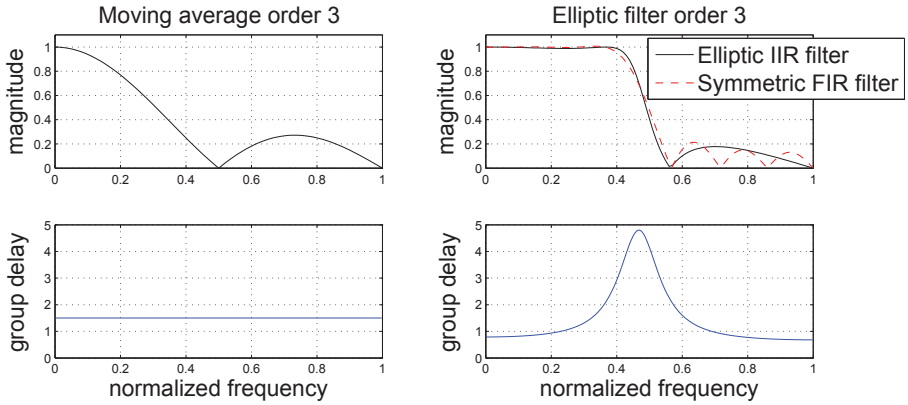


Figure 4.10: The magnitude and group delay response of a moving average filter (left) and an elliptic IIR filter design (right). While the elliptic filter has a much better magnitude response, this comes at the expense of a *non-constant* group delay. The FIR filter has a constant group delay of 1.5 samples for the entire frequency band, while the elliptic filter will give different delays for different frequencies. For example, a signal with a normalized frequency of 0.2 and 0.4 are delayed with about one or three samples, respectively. On the other hand, to get a somewhat similar frequency response as that of the elliptic filter, we need to use a symmetric FIR filter of order 15 (red dashed line in upper-left plot), which gives a constant group delay of 7.5 samples.

difference between FIR and IIR filters. Notice the difference between Figures 4.8 and 4.9.

There are satisfactory design methods for most typical filter types if we mainly consider the magnitude response [32]. However, as already mentioned, we are concerned with not only the magnitude response but also how the design affects the filter delay, i.e. the *group delay*.

4.3.5 The group delay τ

The *group delay* is a measure of how the transfer function, i.e. the filter, delays the signal as a function of the frequency. The typical group delay difference between an FIR filter and an IIR filter is shown in Figure 4.10. A non-constant group delay in the passband will lead to a phase distortion, i.e. different frequency parts of the signal get unevenly delayed. This is known as the *group delay error*, which is also known as *nonlinear phase*. If no phase distortion is wanted in the passband, the group delay needs to be *constant* in the whole passband. And if low filter delay is wanted, the group delay needs to be *low* in the passband. The group delay of a digital filter is given in samples, i.e. sample periods. In other words, a group delay of $\tau = 2$ samples for a system that has a sampling frequency of 100 Hz, yields a time latency of $\tau \cdot f_s = 2 \cdot \frac{1}{100} = 0.02$ seconds (or 20 milliseconds).

4.3.6 Summary of the analysis: The filter objectives

We have now gone through the main important objectives of a digital filter design. First of all, we want the filter to follow some specific magnitude response, either a classic low-pass filter configuration with a certain cutoff frequency or similar low-pass characteristics that follow the ideal differentiator curve. At the same time, low delay is required, which mandates a low group delay in the passband. Finally, we are also interested in a constant group delay, i.e. low phase distortion. To summarize, we want the following:

1. *Low passband distortion*, i.e. a passband that follows the wanted magnitude response in the passband.
2. *High stopband attenuation*, i.e. high noise suppression.
3. *Low group delay*, i.e. low latency.
4. *Constant group delay*, or linear phase.

In this section, we have gathered the needed mathematical expressions, i.e. Equations (4.6) and (4.8), to formulate our filter design challenge. We can rewrite the above objectives into the following error functions:

$$\begin{array}{rcll} err_1 & = & \max(|H(e^{j\pi\omega})| - f(\omega)) & \omega \in [0, \omega_c] \\ err_2 & = & \int_\omega |H(e^{j\pi\omega})|^2 & \omega \in [\omega_c, 1] \\ err_3 & = & \max \tau(\omega) & \omega \in [0, \omega_c] \\ err_4 & = & \max \tau(\omega) - \min \tau(\omega) & \omega \in [0, \omega_c] \end{array}, \tag{4.9}$$

where ω_c represents the cutoff frequency and $f(\omega)$ gives the wanted magnitude response in the passband. The latter was either 1, ω or ω^2 which corresponds to low-pass filters or low-pass differentiators of degree 1 or 2, respectively. These error functions serve two purposes: First of all, they allow us to use automatic design processes by using different optimization algorithms to find the wanted filter behavior. Second, they allow us to make proper comparisons between different filter designs. The latter is the goal of Section 4.6.2. Let us now first go through some of the possible filter design methods.

4.4 Filter design methods

There are several ways of designing digital filters, and the different methods offer different unique trade-offs between the different filter objectives. For brevity, I will not give a full review of the existing range of filter design methods but only mention the most important methods that are relevant for this thesis. Let us start with the design of symmetric FIR filters.

4.4.1 Established symmetric FIR filter design

The design of symmetric FIR filters is a linear problem and there exist different general solutions for most FIR design problems, e.g. the *least square method* and the *Parks-McClellan* method [41, 30]. While the latter solves the filter design problem in the frequency domain in a *max-min* fashion, the former gives a *least mean square* solution. The *least mean square* method is therefore preferable if we want to maximize the noise suppression if the given noise has a white noise distribution [32].

There are other filter design methods that can produce symmetric FIR low-pass filters. One example is *Savitzky-Golay* filters [47]. This filter design method works by choosing a set of filter coefficients that are equivalent to fitting the data to a polynomial around a single input point, i.e. they perform a *local polynomial regression*. By choosing the correct *polynomial order* and *filter length*, it is possible to design filters that preserve the shape and height of waveform peaks. This gives an interesting time domain approach to digital filter design, and the resulting filters can have similar performance to the filter design methods mentioned above. However, the relation to the frequency domain properties is cumbersome. If the frequency domain properties of the data are known, the above standard FIR filter design methods are both more convenient to use and give more filter design possibilities than polynomial fit approaches [47]. Additionally, it is not likely that the polynomial fit approach has any beneficial aspects for filtering MoCap data since human motion does not necessarily follow polynomial curves [42, p. 235]. A comparison between Savitzky-Golay filters with the least mean square method is given in Section 4.7.2.

There are also some examples of *asymmetric FIR* designs, which can give filters with reduced group delay compared with symmetric filters [50]. The design of such filters is a non-linear problem, and there exists no general optimal design method. My results in Paper V also indicate that IIR filters have more low-delay potential than asymmetric FIR filters for a similar computational cost. A low-delay comparison between IIR filters and asymmetric FIR filters is given in Section 4.7.3.

4.4.2 Established IIR filter design methods

Symmetric FIR filters have a fixed group delay of $n/2$, where n is the given filter order. In other words, their constant group delay comes at the expense of a fairly high filter delay compared with IIR filters with similar performance, as we have seen in Figure 4.7. It is therefore relevant to consider IIR filters if high-performance digital filters with low delay are wanted. However, the design of IIR filters is, unlike symmetric FIR filters, a nonlinear problem, and there exist no general optimal design methods. There are different construction methods that can give optimal solutions for some special cases. The most known classical IIR filter methods are

called Butterworth, Bessel[1], Chebychev and elliptic (or Cauer)[67]. They are very useful for standard filter types. However, one typically has little control over the group delay responses [32]. It is therefore necessary to use alternative design methods if more control is needed over the group delay specifications, e.g. if *low* group delay is wanted.

4.4.3 Designing low-pass differentiators

To cascade a standard low-pass filter with some suitable differentiator operator is the most straightforward approach to designing low-pass differentiators. However, this is not necessarily an optimal way, as we have shown in Paper VI. The above general symmetric FIR design methods can design low-pass differentiators with selectable passband and stopband regions [49]. The *firls* method in MATLAB, an implementation of the least square method, offers such functionality. It is also possible to use the Savitzky-Golay method to make low-pass differentiators. However, as explained in Section 4.4.1, this method is limited and cumbersome to use compared with the above general FIR design methods.

I have not found any tools that can design FIR or IIR low-pass differentiators of *degree* 2 with customizable frequency specifications. Using low-pass differentiators of *degree 2 is a more* optimal approach than using a cascade of two low-pass differentiators since we can make a more balanced filter implementation by spreading out the poles and zeros in the z-plane. If we use a cascade of two low-pass differentiators of degree 1, each pole and zero is duplicated in the z-plane. The general designs of IIR low-pass differentiators of *degree 2* that I proposed in Paper VII may be the first presented in the literature.

4.4.4 Filter design through optimization ()

There are several filter design methods in the literature that use different optimization techniques to design alternative IIR filters, given the limitations of the above classical IIR filter designs [10, 55, 50, 46, 7, 32, 34, 61]. These methods typically involve prescribing a desired magnitude and group delay response and transforming the nonlinear IIR filter design problem into a series of linear mathematical programming problems, which then are solved by different numerical methods. However, a common problem with these methods is that the linearization process restricts the designs in different ways [31]. In Paper V, they were also found to not be suitable for our task of minimal delay, since they typically were found to be limited to a lower group delay of $\sim n/2$ [34].

4.4.5 Proposed alternative filter design method: *UR IIR designs*

Since I wanted to explore filter designs with a minimal amount of *group delay*, it was necessary to find an alternative and unrestricted filter design approach. The approach I used was to regard filter design as a *multi-objective optimization problem* [12], which was solved using an *unbiased metaheuristic search algorithm* [44]. The main idea behind this method was to let an algorithm

[1]Bessel is a filter construction method known from the analog world that has a maximally flat group delay response. Bessel filters are seldom used in the digital domain since it is possible to use symmetric FIR filters that have a constant group delay.

find the best possible distribution of poles and zeros directly in the z-plane that optimized a weighted problem based on the filter objectives given in Section 4.3.6.

The proposed method was capable of successfully designing nearly optimal filters with *arbitrary* specifications, including IIR low-pass filters with minimal group delay and IIR low-pass differentiators, as shown in Papers V and VI. In other words, the method was shown to satisfactorily explore unrestricted filters with the wanted trade-off between group delay and the other filter objectives given in Section 4.3.6. Additionally, the method was useful in uncovering the potential of different filter design methods. The method has, given the nature of the heuristic approach, a high computational cost and does not work for high IIR filter orders (> 6). However, as we have shown in Paper VI, it was capable of designing more optimal filters than currently available elsewhere. This and the fact that the method finds designs similar to elliptic designs when magnitude optimal filters are wanted (Paper V) have given credibility to the proposed design method.

The main difference between alternative filter design methods found in the literature and the proposed method is that the latter approach is not based on linearization of the nonlinear filter design problem, which is known to restrict the possible set of solutions [31]. In the following, I will therefore refer to the proposed designs in this thesis as *unrestricted IIR filters*, or *UR IIR filters*. For more details about the design method, see the appendix of this thesis.

4.5 The optimal cutoff frequency when filtering MoCap data

Up to now, I have discussed digital filters and how to design them. Yet I have not discussed how to determine the specifications of the filters, i.e. the requirements of the filter objectives specified in Section 4.3.6. Most of the filter properties are application specific, and it is therefore difficult to discuss these properties in a general way. For instance, the delay specification may be very important for some applications (e.g. rhythmic tasks where high temporal accuracy is wanted), while higher noise attenuation may be more important for other applications (e.g. smooth continuous control tasks when the MoCap data is very noisy). In other words, the given application determines how we should trade off the different filter objectives when designing the most suitable filter. Nevertheless, during the work of this thesis, there was one important filter design challenge of a more general character that caught my attention. If it is reasonable to use the frequency domain approach as a way to separate the motion data from the noise, what is then a sensible cutoff frequency value? Before I discuss ways to determine the frequency content of motion data, let us start by considering the typical noise properties of MoCap systems, i.e. what do we want to filter out?

4.5.1 MoCap noise

There can be many sources of noise in a MoCap system: it can be sensor noise, wobbling markers, electrical interference, quantization noise and more, dependent on the MoCap system used [69]. This noise can be seen as errors, i.e. deviations from the original motion. By adapting suitable filters, we can get better-quality MoCap data since we, in effect, minimize the errors. As already mentioned, sensors, including most MoCap technologies, are known to have white noise properties [66, 69]. This type of noise is evenly distributed in the whole

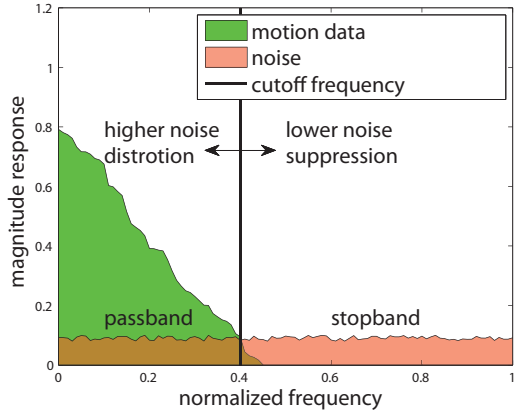


Figure 4.11: The figure shows the dilemma with white noise. It may be necessary to compromise passband distortion by lowering the cutoff frequency inside the passband, to get the desired noise suppression.

frequency band, as illustrated in Figure 4.11. There are exceptions, but this is probably the most reasonable MoCap noise generalization we can make. For simplicity, I will therefore in the following regard the MoCap noise as being white. Consequently, our goal is to attenuate as much as possible of the frequency band that is not part of the wanted signal band, i.e. passband. If it is mandatory to not distort the wanted signal, we need to choose a cutoff frequency that is just outside the passband. However, if we need higher noise suppression than what is possible with the latter conservative choice, we need to compromise signal distortion by lowering the cutoff frequency inside the passband, as illustrated in Figure 4.11. The determination of the optimal cutoff frequency will then be based on the required noise attenuation and how much the frequency cutoff can be lowered inside the passband without excessively distorting the desired signals.

4.5.2 Methods for estimating optimal cutoff frequency

To be able to estimate reasonable cutoff frequencies, I have mainly used two techniques: *power spectral density* (PSD) estimation and a method known as *residual analysis*. Both methods offer a similar view of the frequency content of some given MoCap data. The PSD method gives a frequency spectrum view of the data in power, e.g. decibels (dB), while the residual method gives the root mean square (RMS) distance between the raw and filtered data when using different cutoff frequencies. The latter method was found in Paper VII to be a more robust method. It was also experienced as a more intuitive tool since RMS distance is easier to interpret than power in dB. For instance, if the RMS distance is relatively small for a given cutoff frequency, e.g. < 1 mm for large body motion, it can be seen as neglectable. A reasonable cutoff frequency should first be considered when the *deviation* starts to become problematic for the application. Additionally, I recommend comparing the actual raw data with the filtered data to get a good visualization of how the filters affect the MoCap data. A general implementation of the *residual analysis* method, specifically made to give a frequency analysis of some recorded

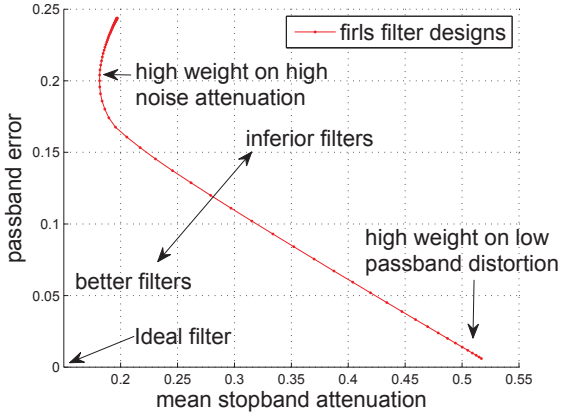


Figure 4.12: The performance of different symmetric low-pass FIR filters of order 4 with a frequency cutoff of $\omega_c = 0.2$. These filters were produced by the *firls* method in MATLAB.

MoCap data, is given in Section 5.3.2.

If we want to filter real-time data, it is necessary to determine such an optimal frequency cutoff beforehand. This was the purpose of an experiment we conducted in Paper VII, where we wanted to find the typical frequency properties for free hand motion. Based on this experiment, we proposed to use cutoff frequencies between 5 and 15 Hz when filtering free hand motion, depending on the type of motion and the needed noise attenuation.

4.6 Low-delay comparison of filter design methods

As shown above, there is a wide range of available filter design methods. When trying to find a suitable filter for a specific application, it is therefore necessary to compare them in a way that makes us capable of choosing the appropriate filter. The purpose of this section is to give a comparison between the low-delay performance of the different filter design methods, based on the error functions given in Section 4.3.6.

4.6.1 Comparison method

A straightforward way of showing the performance of a set of filters is to plot the *noise attenuation* and *passband distortion* for each filter design in one graph. Each dot in the graphs then corresponds to a specific filter design. A good illustration of how this strategy works is to plot the possible set of filters that the symmetric *least mean square* FIR filter design method can produce (the *firls* method in MATLAB). With this method, it is possible to specify the wanted trade-off between the passband distortion and noise attenuation. A good visualization of the possible set of filters that this method can produce can be made by plotting the resulting filter performance for a wide range of different weights, as shown in Figure 4.12. Additionally, this plotting method provides a good way of comparing different filter designs. If we find filters that are closer to an ideal filter, i.e. below the line consisting of *firls* designs, then this implies

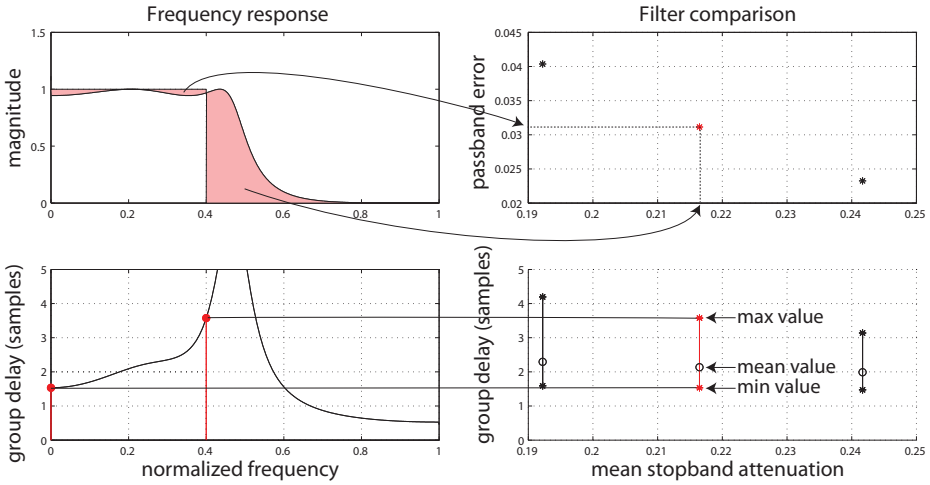


Figure 4.13: Illustration of how the comparison graphs are built. Notice how the four filter performance objectives for the given filter design (left) are reflected in the comparison graph (right) by the three red points.

that we have found *better* filters, i.e. with a better magnitude performance, than the *firls* filters method can produce. Likewise, filters that are found to be above this line are *inferior* filters.

To be able to compare filters thoroughly, we need to use all relevant filter performance criteria. In the following, I will use the four filter objectives specified in Section 4.3.6. In order to visualize the performance involved in these four objectives, it is necessary to use several subgraphs. As long as the sub graphs share the same x-axis and every filter has their unique x-axis values, it is possible to identify the same solution among the graphs since every solution is connected if you draw a vertical line between the graphs. The *x-axis* can be chosen in a way that best identifies the most important performance properties. In the following I have chosen to use the *noise attenuation gain* as the x-axis, since it can be seen as the most important objective. Additionally, since the group delay performance is represented in error functions 3 and 4, I have found it reasonable to plot both objectives in one y-axis. That is, instead of plotting the *maximum group delay* and the *group delay error* in separate axes, I have chosen to plot the *maximum* and *minimum* group delay in one axis. The difference, or height, between the maximum and the minimum then reflects the group delay error, as shown in the example of Figure 4.13. In this plot, it is also possible to mark the mean group delay error to show an overall trend value. In the following, the mean group delay values will be marked as black circles.

4.6.2 Low-delay comparison of filters

I have chosen to compare filters that have a maximum group delay of about two samples in the passband (± 0.02). I found this delay limitation to be a sensible comparison value. First of all, this group delay value was found to produce usable filters with a somewhat balanced trade-off between the different filter objectives. Second, this allows us to use symmetric FIR filters of

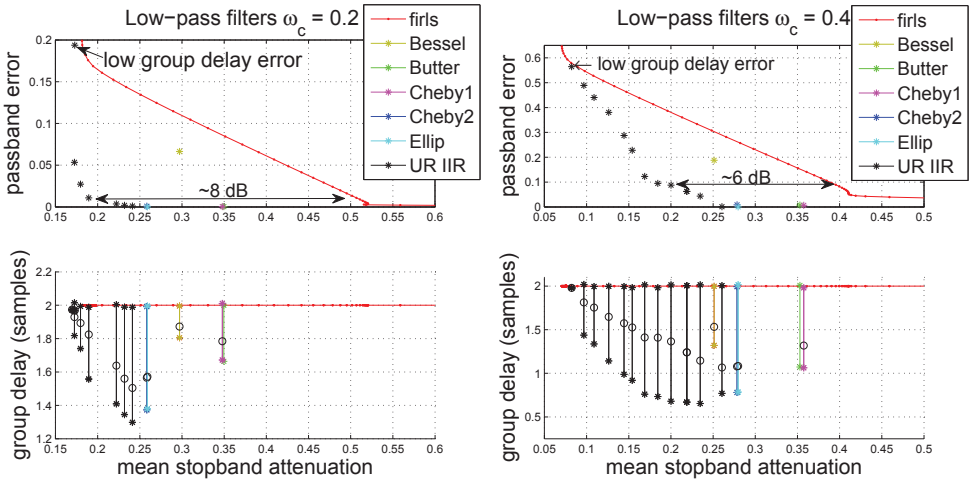


Figure 4.14: Comparison of performance of different low-pass filters of order 4, which have a maximum group delay of roughly two samples in the passband. The normalized cutoff frequency is $\omega_c = 0.2$ and $w_c = 0.4$. The UR IIR designs, designed by the method proposed in this thesis, have more ideal magnitude response. The performance gain comes at the expense of some group delay error compared with symmetric FIR filters. The double arrow reflects the potential noise attenuation gain of using the proposed IIR filter as opposed to the optimal symmetric FIR filters.

order 4, which makes it possible to design low-pass differentiators of degree 1 and 2. I have likewise chosen to compare their performance with IIR filters of order 4. As I have shown in [53], there is not much to gain by increasing the IIR filter order above 4 for this group delay specification. The delay of two samples, which yields a time delay of 20 milliseconds for a 100 Hz MoCap system, may also be in a sensible latency penalty region for real-time musical applications. Notice that a group delay constraint of two samples, may not offer sufficient noise attenuation. If more noise attenuation is needed, it is either necessary to increase the filter delay or to lower the frequency cutoff inside the passband.

Low-pass filters

Figure 4.14 compares the performance of different low-pass filters where the group delay in the passband is limited to about two samples with a normalized cutoff frequency of 0.2 and 0.4. As can be seen in these plots, all IIR filters have better magnitude response, i.e. they are closer to the ideal response, than the symmetric FIR filters. Among the classical IIR filter design methods, Chebychev 2 and elliptic have the best combination of passband distortion and stopband attenuation performance. However, the proposed *UR IIR* filter design method was able to design a range of filters with even better magnitude response. Notice that, unlike our *unrestricted* design approach (UR IIR), the classical IIR filter design methods offer only one solution each given the group delay restriction of $\tau = 2$. This was expected since these methods restrict how the poles and zeros are positioned in the z-plane [37] and was also the main reason for developing the *unrestricted* filter design approach.

While the set of symmetric FIR filters have a constant group delay of two samples, the

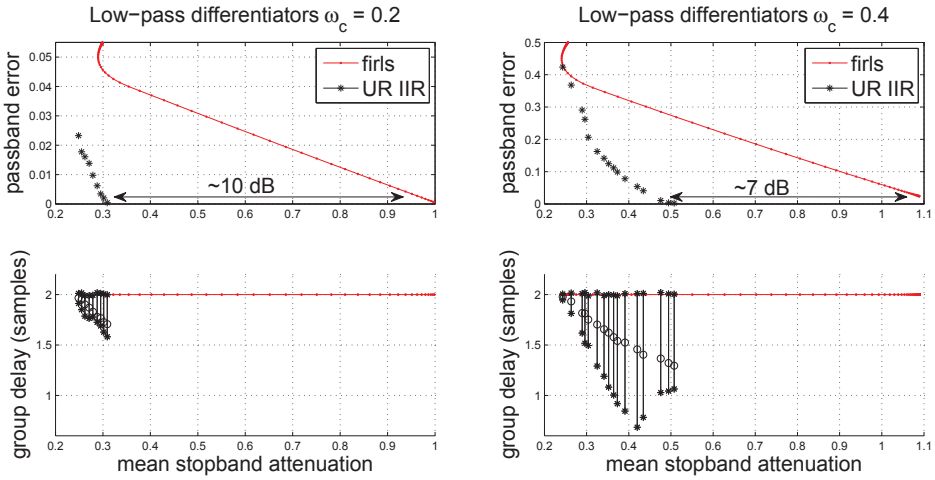


Figure 4.15: Comparison of the performance of different fourth-order low-pass differentiators with a maximum group delay of roughly two samples in the passband, with a frequency cutoff of $\omega_c = 0.2$ (left) and 0.4 (right). The *UR IIR* designs have a far better magnitude response than the symmetric FIR filters, on the expense of some group delay error. The double arrow reflects the potential noise attenuation gain, 7 dB, of using a UR IIR design compared with symmetric FIR filter with the same passband error.

different IIR filters give different group delay errors. It is possible, with the *UR IIR* approach, to design a wide range of IIR filters with different group delay error specifications.

Low-pass differentiators of *degree 1*

When we now continue to compare low-pass differentiators, I continue to compare UR IIR filters against symmetric *firls* designs since they are a good reference (linear phase alternative to IIR filters). I have not included the classical IIR design methods since they give suboptimal designs, as we have shown in Paper VI. To my knowledge, there exist no established IIR design methods that can design non-cascaded low-pass differentiators with customizable passband and stopband.

Figure 4.15 compares the performance of different low-pass differentiators with the same limitation of the group delay value, $\max(\tau) < 2 \pm 0.02$. The UR IIR filter designs show similar performance gain as the low-pass filters above. However, the potential stopband attenuation gain compared with the *firls* design is even greater than for the low-pass filters above. The *group delay error* performance is also better, i.e. lower, than for the above low-pass filters.

Low-pass differentiators of degree 2

Figure 4.16 contains a comparison plot of the performance of different low-pass differentiators of degree 2, with a limited group delay value of two samples. Again, the magnitude performance of the UR IIR filter designs is better than that of symmetric FIR filters[2]. Notice also

[2]Since the *firls* routine does not support the design of *low-pass differentiators of degree 2* directly, I have in the above comparison added an extra zero at dc ($\omega = 0$). This seems to give optimal solutions for the relatively simple

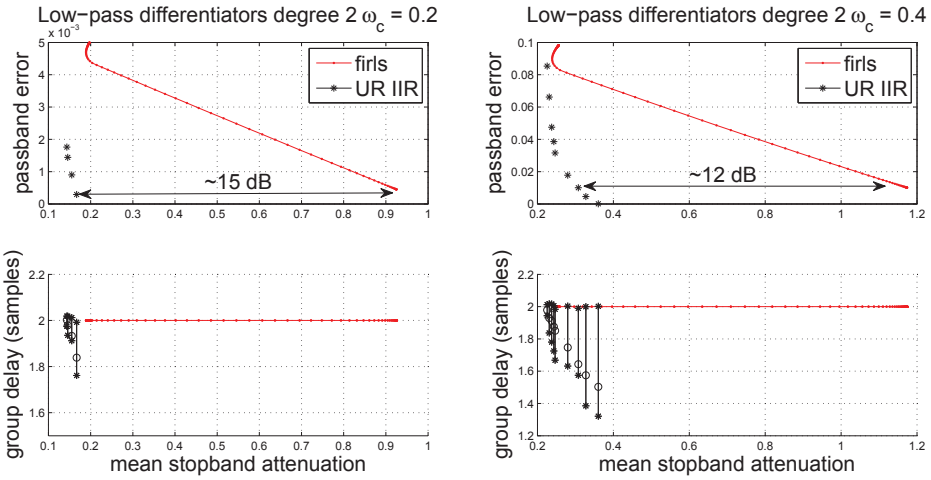


Figure 4.16: Comparison of the performance of different fourth-order low-pass differentiators of degree 2, for cutoff frequencies of 0.2 and 0.4. The *UR IIR* filter designs have a large magnitude performance gain with moderate group delay errors.

the additional improvement of the group delay error compared with *low-pass differentiators of degree 1* above.

Summary of comparison of low delay filters

Overall, based on the comparison plots above and similar plots for other cutoff frequencies, the *UR IIR filters,* which were produced by the proposed filter design method, have the *best magnitude performance* at the expense of some group delay error. Notice also that it is possible to design UR IIR filters with lower passband distortion than what is achievable with the *firls* method. The best performance gain, compared with the *firls* method, is also achieved for filters with low passband distortion. An interesting observation, among the UR IIR solutions, is the gradual increase of the *mean group delay* value with increasing stopband attenuation. This shows a clear trade-off relationship between these two objectives, which coincides with our results in Paper V. Notice also that the group delay error is highly connected to the width of the passband, giving larger group delay errors for wider passbands. It is possible to design UR IIR filters with very low group delay error. However, for large cutoff frequencies ($\omega_c = 0.4$), such filters have a rather poor passband distortion performance and give little improvement compared with the symmetric FIR solutions.

Some of the UR IIR solution sets show some inconsistency in how their performances develop in the comparison plots. Some of the solutions are also partly grouped in clusters. This is probably because of the limitation of how the poles and zeros can be distributed in the z-plane and the restriction in the used search algorithm.

Table 4.1 summarizes the performance gain potential of using the proposed *UR IIR* filters compared with *symmetric FIR filters*. This table is based on the same comparison method that

design problem (it is only necessary to determine the position of one pair of zeros).

Table 4.1: Potential noise attenuation gain in dB of UR IIR filter design compared with optimal symmetric FIR designs (all of order 4).

Normalized cutoff	0.1	0.2	0.3	0.4	0.5
Low-pass filters	**8 dB**	**8 dB**	**8 dB**	**6 dB**	**5 dB**
Low-pass diff. of degree 1	**10 dB**	**10 dB**	**9 dB**	**7 dB**	**6 dB**
Low-pass diff. of degree 2	**16 dB**	**15 dB**	**13 dB**	**12 dB**	**10 dB**

was performed in Figures 4.14, 4.15 and 4.16. As we can see, the highest performance gain is reached for low-pass differentiators of degree 2 with low cutoff frequencies. This is the same table that was presented in Paper VII. To get the same magnitude performance with symmetric FIR filters, it is necessary to use higher-order filters with higher delay penalties, as shown in Figure 4.17.

Probably the most important observation is that the UR IIR approach offers some very interesting and effective designs of low-pass differentiators with *little* group delay error. The best performance gain is achieved for *low-pass differentiators of degree 2* with low cutoff frequencies ($\omega <\sim 0.2$). One example of such a design is shown in Figure 4.18.

However, using the more magnitude-optimal UR IIR filters normally involves some group delay error. It is important then to consider what impact a moderate amount of group delay has on our applications. While it is common to try to obtain constant group delay to ensure an *undistorted* phase, it is possible to imagine cases when this is not of highest importance, e.g. when the low delay performance is more important. (One such example is given in Section 4.7.1.) This will vary for different applications, but some group delay error has a minimal negative effect based on my experience. This is also reasonable, given that human motion lies,

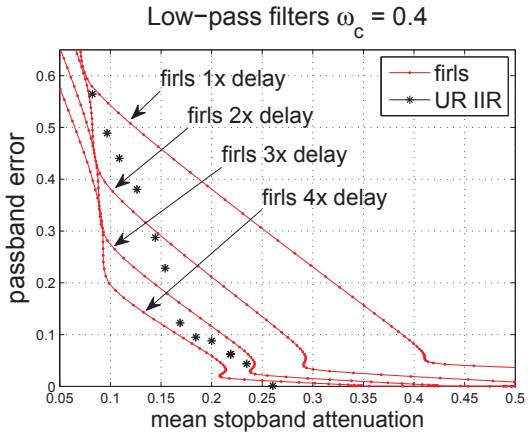


Figure 4.17: Comparison between UR IIR low-pass filters with higher-order *firls* filters. Notice that it is necessary to use *firls* filters with between two to four times the delay, i.e. 4 to 8 samples delay, to get the same magnitude performance as the proposed UR IIR designs. Notice also that for a very low passband error, it is necessary to use even-higher order *firls* filters. However, for large passband errors, less performance is gained with using the UR IIR filters. Similar results were found for different filter specifications.

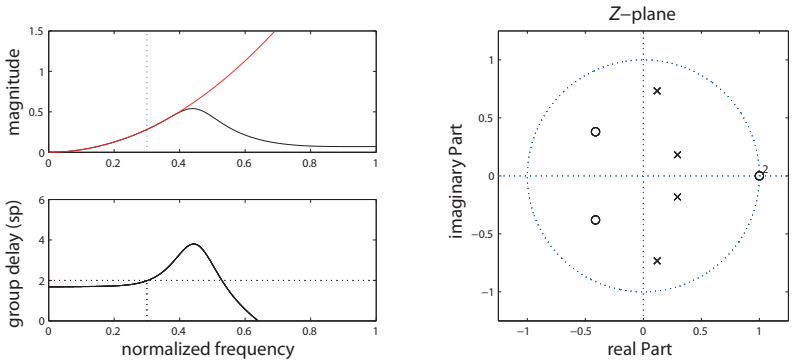


Figure 4.18: Example of a UR IIR design of a low-pass differentiator of degree 2 with a cutoff frequency of $\omega_c = 0.3$. Notice the placement of the two zeros inside the unit circle, which is typical for low-delay designs. The two zeros on the x-axis at $\omega = 0$ are necessary to get the wanted magnitude response of a differentiator of degree 2, i.e. $G(\omega) = \omega^2$. See the appendix for a complete specification of the proposed filters.

for the most part, in the lower part of the frequency spectrum, as shown in Paper VII.

Notice that the comparison above only relates to the theoretical group delay of the given filters. We have not included the computational cost of using the different filter types. However, the cost of running low-order FIR and IIR filters is, in most cases, ignorable given the low sampling frequency of most MoCap systems. In other words, the amount of group delay is much more important than the computational cost of running such filters. A related concern is how asymmetric FIR filters compare with the found IIR filters, which is targeted in Section 4.7.3.

4.7 Additional filter comparisons

4.7.1 Time domain view of differentiators

In this chapter, we have mainly focused on how the filters perform in the frequency domain. However, the filters' behavior in the time domain can be equally important. As discussed in Section 4.3.1, the impulse response reflects how a filter works in the time domain. To be able to low-pass filtering a digital signal, it is necessary to use some filter coefficients that smooth out the noise. The downside of this is that the impulse response will in a similar manner be smoothed out. In other words, the cost of using a low-pass filter can be a trade-off with the resolution in the time domain. The impulse response can, in other words, be a valuable tool to examine the time domain properties of a low-pass filter.

To get a similar response of low-pass differentiators, the impulse response is not necessarily correct, since the impulse response will show both the differentiator process together with the low-pass filter process. It is therefore necessary to use the integral of impulse response, i.e. the *step response*. Likewise, to see a similar response of a *differentiator of degree* 2, it is necessary to use the *double integral* of the impulse response, i.e. the *ramp response*, as shown in Figure 4.19. Limb collisions, e.g. a hand clap, can be interpreted as objects having a constant velocity

that are brought to a sudden stop by a collision, which resembles a ramp response in reverse. In other words, the ramp response shows how the positional data of an ideal collision is treated by a double differentiator, i.e. what we get for acceleration data. In the Dance Jockey project, we actively used such limb collisions and their acceleration data to trigger samples and musical features. Maximizing the recognition rate of such collisions is therefore a relevant challenge.

In Figure 4.20, the *absolute* ramp responses of four different double differentiators are shown. While using two finite difference equations in cascade gives the ideal ramp response in the time domain, it offers little noise suppression. By cascading this filter with a moving average filter of order 2 (length 3), we get some more noise suppression. However, the energy of the collision is equally spread out in three samples, which is not necessarily good if we want to detect the collision among noisy data. A similar effect was discovered in Paper VII. Since a collision can be regarded as having a flat spectrum, it is necessary to include some of the frequency band to be able to detect collision among noisy data. In other words, using moving average filters can easily remove too much of the energy of a collision, making it harder to detect the collision among noisy data.

The absolute ramp response of the *low-pass differentiator design* with the *firls* method gives a much more easily detectable spike. The absolute ramp response of the *UR IIR low-pass differentiator of degree 2* has a similar spike and delay as the *firls* design, yet with more noise suppression. Additionally, the *UR IIR* filter has a much lower passband distortion than the *firls* design, which is not reflected by the ramp response. The improved performance of the *UR IIR filter* design is at the expense of some group delay error, which is reflected by the extra tail in the ramp response (bottom-left graph in Figure 4.20). However, such a group delay error does not necessarily have any negative impact on the recognition of a limb collision, as shown by the bottom-right graph in Figure 4.20. Indeed, in this example, the UR IIR low-pass differentiator gives the best ratio between the peak collision value and peak noise value, which relates to the most easily detectable collision.

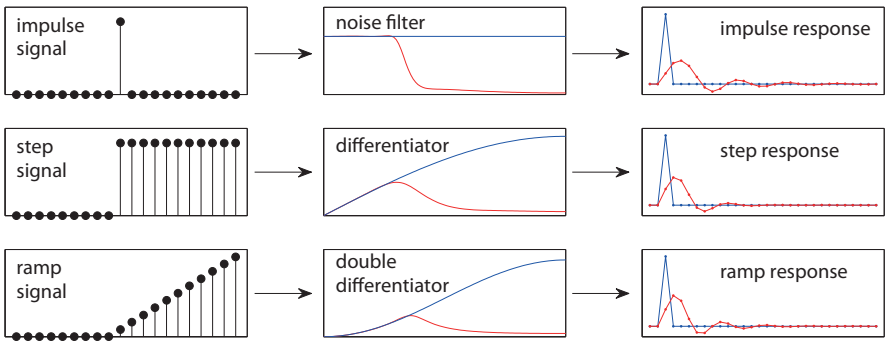


Figure 4.19: Illustration of the how the *impulse*, *step* and *ramp* responses give a similar time domain view *of low-pass filters* and low-pass differentiators of degrees 1 and 2, respectively. The blue lines correspond to the non-smoothed versions of the filters, i.e. *no* filter or finite difference equations, while the red lines correspond to the proposed UR IIR filter designs. The step and ramp responses take away the derivative processing element of the differentiators and show only the remaining low-pass filter coefficients. Notice how similar the different responses are since they have similar low-pass configurations with a normalized cutoff of around $\omega_c = 0.3$.

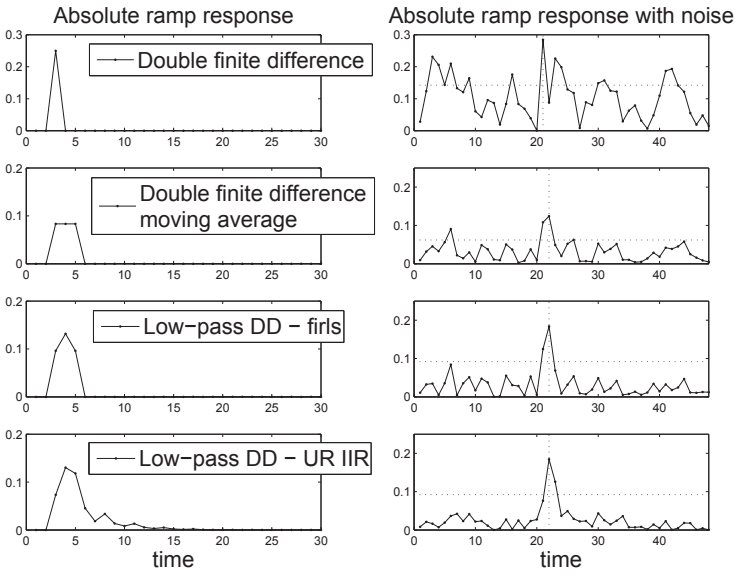


Figure 4.20: Absolute ramp response for different double differentiators, without noise (left) and with noise (right). Notice that the *UR IIR* design has the best noise suppression and that group delay error is not problematic if the task is to recognize a collision peak among noisy data. All differentiators were fed with an identical noisy ramp signal. The horizontal stippled line gives half of the maximum value in the current graph and reflects how good the differentiators are to distinguish a collision from the surrounding noise, similar to the concept of *signal-to-noise ratio*. Notice that the three lower filters have similar filter delays.

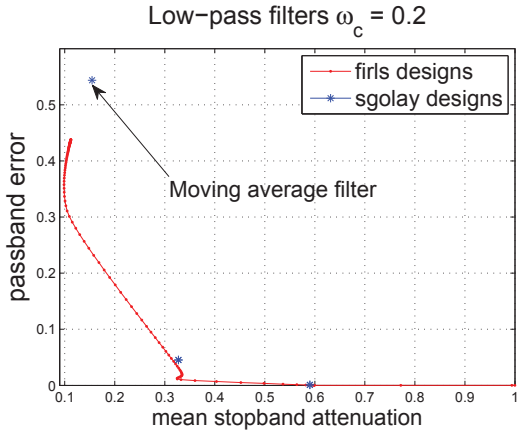


Figure 4.21: Comparison between symmetric FIR filters or order 6 made by the *firls* method and the Savitzky-Golay method. Notice that since these symmetric FIR filters are of the same order, they have the same constant group delay of three samples.

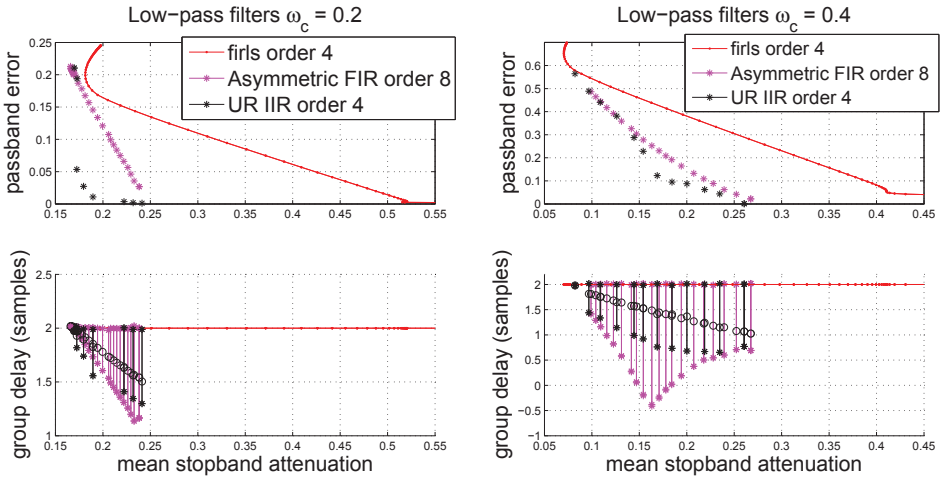


Figure 4.22: Comparison between UR IIR filters of order 4 and (unrestricted) asymmetric FIR filters of order 8, all with a maximum group delay of roughly two samples and a frequency cutoff of $\omega_c = 0.2$ and 0.4. The *firls* solutions in red are given as reference.

4.7.2 *Savitzky-Golay* versus the *least square* method

We can use the same comparison method from Section 4.6.1 to compare the *least square method (firls)* with the *Savitzky-Golay* method. This is done in Figure 4.21. The Savitzky-Golay method can only produce three different filters when the filter order is set to be 6, and the first is equivalent to a moving average filter. While some of the Savitzky-Golay filters have similar performance as the *firls* designs, the *firls* method gives much more design possibilities [47], as shown in Figure 4.21. The extra design possibilities for the same filter order should be beneficial for most applications. Additionally, it is much easier to design suitable *firls* filters if the frequency properties of the wanted filter is known.

4.7.3 *Asymmetric FIR* versus *UR IIR* filters

Figure 4.22 shows a comparison plot between *UR IIR* filters of order 4 and *asymmetric FIR* filters of order 8, both with an upper group delay restriction of two samples. The *asymmetric FIR* filters were found by our alternative filter design method, i.e. the filters cannot be guaranteed to be optimal. Yet the found filters should give a good indication of the expected performance of using asymmetric FIR filters for this task, especially given the consistent results. Notice that asymmetric FIR filters do not have constant group delay.

As shown in Figure 4.22, the found UR IIR filters of order 4 are more optimal than the found unrestricted *asymmetric FIR* filters of order 8. The UR IIR filters have better combination of low-passband distortion and high noise attenuation. IIR filters of order 4 and FIR filters of order 8 can be said to have similar degrees of computational cost.[3] These results coincide with our

[3]From Equations (4.3) and (4.4) we can deduce that FIR filters need n additions and $n + 1$ multiplications, while IIR filters need $2n$ additions and $2(n + 1)$ multiplications, where n is the given filter order. In other words, IIR filters demand twice as many operations as FIR filters of the same order do.

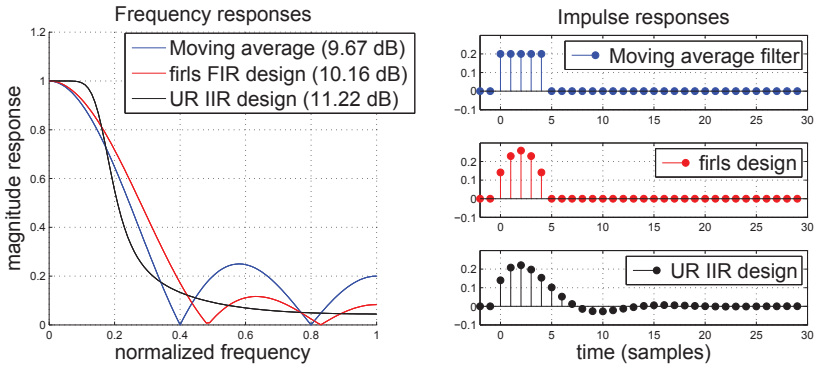


Figure 4.23: Comparison of the maximum noise suppression of filters of order 4 when only demanding $G(0) = 1$. All filters have a group delay of two samples for $\omega = 0$. The impulse responses of each filter are also given (right). Notice the long impulse response of the UR IIR filter, which gives the extra noise attenuation.

results from Paper V. According to my experience, it is necessary to use asymmetric FIR filters of an order between 20 and 30 to copy the performance of UR IIR filters of order 4 proposed in Paper VII. This is not surprising given the recursive structure of IIR filters, which makes them more effective in producing long impulse responses. In other words, *UR IIR* filters seem to be more low-delay optimal than using asymmetric FIR filters with a similar computational cost.

4.7.4 Reducing random noise

The moving average filter is actually optimal for one thing, which is reducing random noise while retaining a sharp step response [54]. This is reasonable given the structure of moving average filters. If the noise is random, none of the input points are special. In other words, there is little sense *weighting* some of the points more or less in order to get more noise suppression. Nevertheless, if we do not need a sharp step response, it is possible to gain some more noise suppression by using the *firls* method or UR IIR filters, as shown in Figure 4.23. While there is not much to gain, the *firls* method with the same filter order gives 0.49 dB additional noise attenuation. An UR IIR filter design of order 4, designed by the proposed filter design method, gives another 1.06 dB noise attenuation improvement with no ripples in the stopband, which may be beneficial for some applications. Notice that the UR IIR filter design is not achievable with the established filter design methods since the zeros need to be inside the unit circle. The specification of these filters is given in the appendix. According to the given results in this chapter, it seems that UR IIR filters always give a better magnitude response when the group delay is restricted, on the expense on some group delay error.

4.8 Discussion and summary

The goal of this chapter has been to review and suggest some best practices for filtering MoCap data for real-time applications. To target these challenges, I have given some backgrounds to digital filters, filter analysis and filter design methods. Given the convincing results from our

experiment of finding the optimal cutoff frequency for filtering free hand motion in Paper VII, I recommend regarding MoCap data in the frequency domain since it seems to be an effective way of separating the motion and noise. A polynomial fit approach does not seem to provide any advantages for general MoCap data.

Using *symmetric FIR* filters is a sensible choice for post-processing use, especially given the constant group delay error and the good availability of different design methods. Furthermore, I recommend using the least square method, e.g. the *firls* method in MATLAB, since it gives optimal symmetric FIR noise filters when the given noise is random or white.

When differentiating MoCap data, I recommend using low-pass differentiators since they avoid the undesirable amplification of the noise in the higher frequencies. Such low-pass differentiators can be designed with the above recommended symmetric FIR design method (*firls*). However, symmetric FIR filters are not necessarily optimal if the lowest filter delay is needed.

No publications that directly targeted the topic of best practices for designing filters with minimal group delay, i.e. low latency, were found. IIR filters seemed like a sensible approach since it is known that the recursive approach offers an effective way of achieving a long *impulse response* without having to use long FIR filters. In spite of this, the established IIR filter design methods were not found to be suitable for designing optimal low-delay filters. I have therefore proposed an alternative filter design method based on multi-objective optimization, which enables more optimal designs of low delay filters than the established methods can produce, as presented in Paper V. I have referred to these designs as *unrestricted IIR* (UR IIR) filters. With this method, I could also design UR IIR low-pass differentiators, which were favorable compared with designs given in the literature, as presented in Paper VI.

To be able to compare the delay performance of different filter design methods, I have shown how these filter design methods compare when the group delay was limited to a maximum of two samples. According to the results presented , there is a lot to gain compared with symmetric FIR filters on the cost of some group delay error. The greatest potential was shown among low-pass differentiators of degree 1 and 2. Compared with optimal symmetric FIR filters, they give a noise attenuation increase between 5 and 16 dB with similar delay, or up to two and four times the delay reduction for similar magnitude properties. Such delay savings can be important for achieving good responsiveness in musical applications.

Finally, in the end of this chapter, I have given some additional filter design comparisons. The results indicate that *UR IIR* filters are more low-delay optimal than asymmetric FIR filters for a similar computational cost. Additionally, it is shown how the ramp response can give a valuable time domain view of how low-pass differentiators of degree 2 process limb collisions.

Chapter 5

Research Contribution

In the following I will give a summary of the papers included in this thesis. The overall content of the included papers is outlined in Section 5.1. An overview of performed Dance Jockey performances and some software that have been made available to others then follow.

5.1 Overview of the included papers

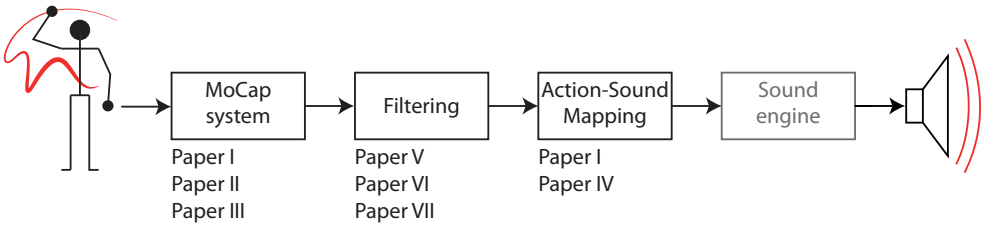


Figure 5.1: How the included papers relate to the challenges targeted in this thesis.

The research conducted in this thesis can be divided into two main parts. In the first part, which includes Paper I to Paper IV, I did research on how to use MoCap technologies for real-time musical interactions and their suitability for such tasks. During this period, I also worked with the Dance Jockey project, where we used the Xsens MVN suit for musical interaction, with which we had several public performances. The period ends with Paper IV, which presents the details of the development of the Dance Jockey system.

During the work of this thesis, I have studied best practices in filtering MoCap data for real-time applications. Since the literature didn't present satisfying answers, an investigation was undertaken. The main result of this work is presented in the last three papers. In Paper V, I presented work that dealt with optimal designs of *low-delay filters,* and in Paper VI, I presented work that dealt with optimal designs of *IIR low-pass differentiators*. Finally, in Paper VII, I summarized my work concerning best practices for filtering real-time data and applied it to the application targeted in this thesis. Let us now take a closer look at the content and motivation of each of the individual papers.

5.2 Papers

5.2.1 Paper I

Using IR Optical Marker Based Motion Capture for Exploring Musical Interaction

This work started with a curiosity of how motion capture technologies could be used for musical interaction. It was therefore relevant to investigate what others had done before and to consider some of the possibilities these technologies had to offer. Since infrared marker based MoCap systems appeared to be one of the best performance systems available, I was interested to see how people had used such systems in the field of musical interaction. In this paper, I also reviewed and tried to conceptualize how such MoCap data could be used to control sound and sonic features. Little work involving the use of full-body MoCap data for real-time musical interaction was found.

Abstract

The paper presents a conceptual overview of how optical infrared marker based motion capture systems (IrMoCap) can be used in musical interaction. First we present a review of related work of using IrMoCap for musical control. This is followed by a discussion of possible features which can be exploited. Finally, the question of mapping movement features to sound features is presented and discussed.

5.2.2 Paper II

OSC Implementation and Evaluation of the Xsens MVN Suit

In the beginning of summer 2010, I started to work with a MoCap system known as the *Xsens MVN suit*. I saw it as especially relevant to develop a robust implementation of the Open Sound Control (OSC) protocol to be able to easily integrate it with different applications and sound engines. During this work, I gained experience on how to use the suit for musical interaction and discovered some problems with the Xsens system. This led me to develop new versions of the OSC implementation to bypass the problems and incorporate new features. Another important subject was to try to quantify the positional tracking performance and the real-time performance of this MoCap system to better understand its strengths and weaknesses when used for controlling sonic and musical features.

Abstract

The paper presents research about implementing a full body inertial motion capture system, the Xsens MVN suit, for musical interaction. Three different approaches for streaming real time and prerecorded motion capture data with Open Sound Control have been implemented. Furthermore, we present technical performance details and our experience with the motion capture system in realistic practice.

5.2.3 Paper III

Comparing Inertial and Optical MoCap Technologies for Synthesis Control

In Paper II, I gave an evaluation of the performance of the Xsens MVN suit. However, there were several other MoCap systems available and it was of interest to understand more about the difference between them. Kristian Nymoen and I therefore started a systematic testing of two such systems; the *NaturalPoint OptiTrack*, an optical marker based system, and the already mentioned *Xsens MVN suit*. We performed several simultaneous recordings with both systems to get a clearer image of the strengths and weaknesses of these systems. Additionally, to get a user's view of the compared technologies, we recruited a musician that was given certain musical-related tasks that he needed to perform with both systems. The recordings done from these tasks and the verbal feedback from the musician were then used in the comparison.

The importance of identifying the real-time performance of such systems was one of the most important points learned during this work. Though the OptiTrack is superior when it comes to accurate positional data, we identified several reasons why Xsens was more suitable as a real-time device and a more robust system when used on stage. For instance, the Xsens system has no occlusion problems and hence offers more consistent and smooth real-time data. Additionally, the Xsens system offers acceleration data directly with little noise problems. We also identified *occlusion noise*, which is a prominent problem with optical marker-based systems. While occlusion noise only contributes to positional displacement of spikes up to about 1 millimeter, such errors get heavily amplified when differentiated and were found problematic in our experiments. These discoveries contributed to my motivation for investigating best practices for filtering MoCap data.

Abstract

This paper compares the use of two different technologies for controlling sound synthesis in real time: the infrared marker-based motion capture system OptiTrack and Xsens MVN, an inertial sensor-based motion capture suit. We present various quantitative comparisons between the data from the two systems and results from an experiment where a musician performed simple musical tasks with the two systems. Both systems are found to have their strengths and weaknesses, which we will present and discuss.

5.2.4 Paper IV

Developing the Dance Jockey System for Musical Interaction with the Xsens MVN Suit

In the end of the summer of 2010 I started the *Dance Jockey project* with Yago de Quay, who at that time was a visiting researcher in our lab. The project was based on my OSC implementation of the Xsens MVN suit. The main motivation behind this project was to use full body motion for musical interaction, where all aspects of the performance should be controlled solely through the Xsens MVN suit. Our goal was to develop a performance piece in which properties of the output sound would match properties of the performed actions. In this way, we wanted to obtain a more physically engaging, communicative, and audience-friendly instrument choreography, as an alternative to the typical laptop performance with which electronic

music is often associated. After only a few weeks of work, we had our first performance at the Department of Musicology in Oslo, Norway. Since we got positive feedback from the audience and found the work exiting, we continued working with the project, which resulted in several public performances in Norway and Portugal. Later, in 2011, our performance was accepted as a part of the concert program of NIME, a conference dedicated to new interfaces for musical expressions. After occasionally working with the Dance Jockey project for over a year, it was time to document the details and our experience of developing the *Dance Jockey system,* which resulted in this paper.

Abstract

In this paper we present the Dance Jockey System, a system developed for using a full body inertial motion capture suit (Xsens MVN) in music/dance performances. We present different strategies for extracting relevant postures and actions from the continuous data, and how these postures and actions can be used to control sonic and musical features. The system has been used in several public performances, and we believe it has great potential for further exploration. However, to overcome the current practical and technical challenges when working with the system, it is important to further refine tools and software in order to facilitate making of new performance pieces.

5.2.5 Paper V

Digital IIR Filters with Minimal Group Delay for Real-Time Applications

When dealing with filters and real-time applications, there is especially one important challenge that I wanted to target: which digital filters minimize the delay they introduce to the system? I suspected that IIR filters would have the best potential for such low-delay designs since they are known to give more effective filter designs given their recursive structure. Yet I could not find any work that answered my questions. This eventually led me to the implementation of an alternative filter design method based on multi-objective optimization combined with a metaheuristic search algorithm. With this method, I was able to design more optimal low-delay filters than currently achievable with the established filter design methods. The method was also able to uncover the potential of using different filter design methods for low-delay designs. This made it possible to present a thorough low-delay comparison between different filter design methods. Additionally, the experimental results suggested a linear relationship between stopband attenuation and the filter delay, giving an upper bound for the achievable noise attenuation for a given delay.

Abstract

In this paper we examine the potential for designing digital (IIR) filters with minimal group delay, which are relevant for real-time applications. By formulating filter design as a multi-objective optimization problem and approaching it with an unbiased metaheuristic search algorithm, we have established relationships between filter delay and other filter objectives. These

relationships are presented as non-inferior surfaces for different filter orders and design approaches. We present possible designs that are realizable with (1) classical IIR design constructions, and (2) unconstrained global search for filter orders between 2 and 5. Elliptical (Cauer) filters are found as to have the highest potential for low group delay among the classical constructions. However, as one might expect, unconstrained IIR search discovers more optimal filters, but is limited to filter orders of 5. Currently, there exists no established method that can construct similar IIR filters with a group delay below n/2, where n is the given filter order. Finally, we present some unconstrained filter examples that we claim are nearly optimal.

5.2.6 Paper VI

Designing Digital IIR Low-Pass Differentiators with Multi-objective Optimization

Best practices for differentiating MoCap data was another filter design challenge I was concerned with. Such operators are frequently used to compute velocity and acceleration data from positional MoCap data. In our labs, we were using the finite difference equation in combination with different low-pass filters to manage the increased noise problem that the former created. However, I suspected that there were more optimal solutions to our differentiation needs. I soon discovered so-called low-pass differentiators that avoid the undesirable amplification of noise in the higher-frequency band, which is characteristic of MoCap data. This is in general a more optimal approach than using a cascade of a differentiator operator and a low-pass filter. Yet I could not find any established design methods for designing IIR low-pass differentiators. Fortunately, since such low-pass differentiators can be seen as a filter in the frequency domain, I could use a similar filter design method that I had been developing for Paper V to design low-pass differentiators with arbitrary specifications. It was encouraging that the proposed design method found IIR low-pass differentiators that compared favorably with designs given in the literature, which gave credibility to the developed design method.

Abstract

In this paper we examine the possibility of designing IIR low-pass differentiators by approaching it as a weighted multi-objective optimization problem and solving it with an unbiased metaheuristic search algorithm. By collecting several solutions with different sets of weights we are able to make a thorough comparison of different design strategies. We present possible designs that are realizable with (1) cascading classical IIR low-pass filters with appropriate operators, and (2) non-cascaded general IIR differentiator designs. Elliptical filters are found to be the most magnitude-optimal among the first type. However, the non-cascaded approach found more optimal IIR differentiators at the expense of a more complicated search. Finally, we present some non-cascaded general designs that compare favorably with the available designs given in literature, and which we reason are nearly optimal.

5.2.7 Paper VII

Filtering Motion Capture Data for Real-Time Applications

After working extensively with developing new tools for designing low-pass filters and low-pass differentiators, it was time to apply the gained knowledge to MoCap data and the applications targeted in this thesis. My main goal was to propose a range of filters suitable for real-time MoCap applications. To be able to design such specific filters, it was necessary to find the typical frequency content of MoCap data that we wanted to filter. The solution I found was to conduct an experiment to find the typical frequency content of free hand motion. The experiments' results showed that it is a useful approach to separate the motion from noisy MoCap data in the frequency domain. Then based on these results, I could start designing a range of filters suitable for real-time MoCap applications. In addition to presenting low-delay IIR low-pass filters and IIR low-pass differentiators, I also presented *IIR low-pass differentiators of degree* 2. It is more optimal to use the latter design than to use two differentiators in cascade. Once again, I could use the proposed alternative filter design method to design the wanted novel filters. I have not found general designs of IIR low-pass differentiators of degree 2 in the literature, so these may be the first presented. Given the large amount of work behind these results, it was necessary to skip several details when writing Paper VII. Some additional details are therefore given in Chapter 4 of this thesis.

Abstract

In this paper we present some custom designed filters for real-time motion capture applications. Our target application is motion controllers, i.e. systems that interpret hand motion for musical interaction. In earlier research we found effective methods to design nearly optimal filters for real-time applications. However, to be able to design suitable filters for our target application, it is necessary to establish the typical frequency content of the motion capture data we want to filter. This will again allow us to determine a reasonable cutoff frequency for the filters. We have therefore conducted an experiment in which we recorded the hand motion of 20 subjects. The frequency spectra of these data together with a method similar to the residual analysis method were then used to determine reasonable cutoff frequencies. Based on this experiment, we propose three cutoff frequencies for different scenarios and filtering needs: 5, 10 and 15 Hz, which correspond to heavy, medium and light filtering, respectively. Finally, we propose a range of real-time filters applicable to motion controllers. In particular, low-pass filters and low-pass differentiators of degrees one and two, which in our experience are the most useful filters for our target application.

5.3 Additional contributions

5.3.1 Dance Jockey performances

We have performed several public *Dance Jockey* concerts during the period 2010–2011. These concerts are listed below in chronological order. Several of the performances are documented

with videos on our project page.[1]

- Department of Musicology, Oslo, Norway (August 25, 2010)
- *Gabler* (dance club venue), Oslo, Norway (Video N.A.)
- VERDIKT Conference, Oslo, Norway (November 1, 2010)
- Mostra UP, Porto, Portugal, (March 18-19, 2011) (two concerts)
- NIME Conference 2011, Chateau Neuf, Oslo, Norway (June 1 , 2011)
- Idefestivalen, Oslo, Norway (September 17, 2011)

5.3.2 Software and tools made available

Various software and tools have been developed during the work of this thesis. In the following section, I will present a subset of these, which I see as relevant to others. The software is available on the software web page of fourMs labs, if not otherwise stated.[2]

Frequency analysis of MoCap data with the residual analysis

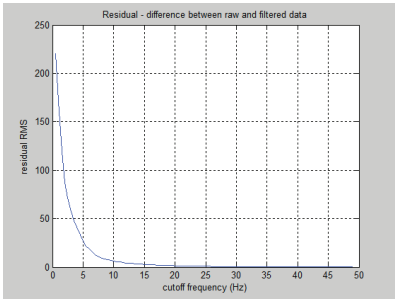


Figure 5.2: Residual analysis plot of a MoCap recording of hand motion.

In Paper VII, we performed an experiment to determine the frequency properties of free hand motion. To analyze the data, we implemented a general form of the residual analysis to be able to determine the frequency content of MoCap data (or similar data). The method consists of low-pass filtering the data with different cutoff frequencies and calculating the residual, i.e. what is left over when we subtract the filtered data from the raw data. As long as the filter is only attenuating noise, the residual should be rather small. However, when the filter starts to attenuate the desired signal, the residual will become larger. By performing this analysis for several cutoff frequencies and plotting the resulting residuals, we get an overall picture of their impact. This plot can then serve as a basis for determining a reasonable cutoff frequency. The function is written in MATLAB and should work for most MATLAB versions.

[1] http://www.fourms.uio.no/projects/sma/subprojects/dancejockey/
[2] http://www.fourms.uio.no/downloads/

Max IIR MoCap filter patch

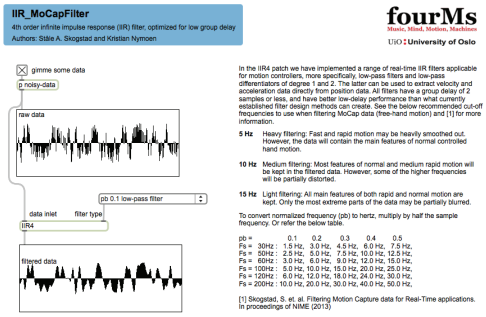


Figure 5.3: Screen shot from the help patch.

A range of near-optimal low-delay IIR filters, proposed in Paper VII, are embedded in a Max patch for easy access. All filters have a group delay of two samples or less and have, to my knowledge, better low-delay performance than what currently established filter design methods can create. The filters, consisting of low-pass filters and low-pass differentiators of degrees 1 and 2, are specified with different normalized cutoff frequencies. To choose a suitable cutoff frequency, see the guidelines in Paper VII or use the above proposed residual analysis method. The specification of the proposed filters is given in the appendix.

OSC implementation of the Xsens MVN suit

During the work of Paper II, I developed three different OSC implementations of the Xsens MVN suit. The first one was a simple JavaScript for Max which had several limitations. Yet the implementation is straightforward and can still be useful for some applications (*Xsens MVN datagram unpacker*). The final and preferable implementation, as discussed in Paper II, was based on the Xsens Software Development Kit (SDK). Due to copyright issues, this implementation cannot be published. However, users that have their own Xsens SDK license can get this implementation on demand.

5.4 List of publications

Papers included in this Thesis

I Using IR Optical Marker Based Motion Capture for Exploring Musical Interaction.
S.A. Skogstad, A.R. Jensenius and K. Nymoen
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 407–410 Sydney University of Technology 2010.

II OSC Implementation and Evaluation of the Xsens MVN suit.
S.A. Skogstad, K. Nymoen, Y.d. Quay and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 300–303, University of Oslo 2011.

III Comparing Inertial and Optical MoCap Technologies for Synthesis Control.
S.A. Skogstad, K. Nymoen, and M.E. Høvin.
In *Proceedings of SMC 2011 8th Sound and Music Computing Conference "Creativity rethinks science"*, pages 421–426, Padova University Press 2011.

IV Developing the Dance Jockey System for Musical Interaction with the Xsens MVN suit.
S.A. Skogstad, K. Nymoen, Y.d. Quay and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 226–229, University of Michigan 2012.

V Digital IIR Filters With Minimal Group Delay for Real-Time Applications.
S.A. Skogstad, S. Holm and M.E. Høvin.
In *IEEE The International Conference on Engineering and Technology. 2012.*, pages 1–6, German University in Cairo 2012.

VI Designing Digital IIR Low-Pass Differentiators With Multi-Objective Optimization.
S.A. Skogstad, S. Holm and M.E. Høvin.
In *IEEE 11th International Conference on Signal Processing. 2012.*, pages 10–15, Beijing Jiaotong University 2012.

VII Filtering Motion Capture Data for Real-Time Applications.
S.A. Skogstad, K. Nymoen, S. Holm, M.E. Høvin and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 196–197, Kaist University, Daejeon 2013.

Other papers

- de Quay, Yago; Skogstad, Ståle Andreas van Dorp; Jensenius, Alexander Refsum.*Dance Jockey: Performing Electronic Music by Dancing*. In *Leonardo Music Journal, 21: 11–12*, 2011.
- Godøy, Rolf Inge; Jensenius, Alexander Refsum; Voldsund, Arve; Glette, Kyrre Harald; Høvin, Mats Erling; Nymoen, Kristian; Skogstad, Ståle Andreas van Dorp; Tørresen, Jim. Classifying Music-Related Actions. In *Proceedings of the ICMPC-ESCOM 2012 Joint*

Conference: 12th Biennial International Conference for Music Perception and Cognition, pp. 352-357*,* Thessaloniki, Greece, 2012.

- Nymoen, Kristian; Jensenius, Alexander Refsum; Tørresen, Jim; Glette, Kyrre Harald; Skogstad, Ståle Andreas van Dorp. Searching for Cross-Individual Relationships between Sound and Movement Features using an SVM Classifier. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 259–267, Sydney University of Technology 2010.

- Nymoen, Kristian; Skogstad, Ståle Andreas van Dorp; Jensenius, Alexander Refsum. SoundSaber - A Motion Capture Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 312–315, University of Oslo, 2011.

- Nymoen, Kristian; Voldsund, Arve; Skogstad, Ståle Andreas van Dorp; Jensenius, Alexander Refsum; Tørresen, Jim. Comparing Motion Data from an iPod Touch to a High-End Optical Infrared Marker-Based Motion Capture System. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 88–91, University of Michigan 2012.

- Jensenius, Alexander Refsum; Glette, Kyrre Harald; Godøy, Rolf Inge; Høvin, Mats Erling; Nymoen, Kristian; Skogstad, Ståle Andreas van Dorp; Tørresen, Jim. fourMs, University of Oslo – Lab Report. In *Proceedings of the 2010 International Computer Music Conference*, pages 290–293, New York, 2010.

- Jensenius, Alexander Refsum; Nymoen, Kristian; Skogstad, Ståle Andreas van Dorp; Voldsund, Arve. A Study of the Noise-Level in Two Infrared Marker-Based Motion Capture Systems. In *Proceedings of the 9th Sound and Music Computing Conference*. pp. 258–263. Aalborg University, Copenhagen, 2012.

- Jensenius, Alexander Refsum; Skogstad, Ståle Andreas van Dorp; Nymoen, Kristian; Godøy, Rolf Inge; Tørresen, Jim; Høvin, Mats Erling. Reduced displays of multidimensional motion capture data sets of musical performance. In *Proceedings of ESCOM 2009: 7th Triennial Conference of the European Society for the Cognitive Sciences of Music,* Jyväskylä, Finland, 2009.

Chapter 6

Summary and Conclusion

This chapter gives a summary of this thesis. Then a conclusion is given, and a direction for possible future work is suggested.

6.1 Summary

The primary research objective of this thesis was to *develop methods and technologies for using body motion for real-time musical interaction.* The work consisted of (1) doing quantitative evaluation of MoCap technologies , (2) developing the Dance Jockey system and (3) studying best practices for filtering real-time MoCap data. These three subcategories coincide with the subobjectives that were presented as the aim for this thesis in Section 1.4. It is therefore reasonable to divide the summary of this thesis into these three subcategories.

6.1.1 Evaluation of motion capture technologies

When using MoCap data to control sonic and musical features, it is obvious that the quality of the MoCap system can influence the performance. It has therefore been important to evaluate the performance of available MoCap systems. The evaluation that was undertaken in Paper III has shown that the two evaluated systems had their different strengths and weaknesses. Additionally, my brief review of the available MoCap system in Chapter 3 indicates the same tendency. There is no single MoCap technology that will fulfill every need. Instead, every available technology offers its different properties with strengths and weaknesses. In order to make reasonable MoCap technology choices, it is therefore necessary to regard the needed performance for the intended application. In this respect, I have found it useful to group the MoCap performance in three main categories:

Data quality. The term data quality is used to refer to the spatial accuracy and precision of the MoCap data output. Our evaluation in Paper III has shown that OptiTrack, an optical marker-based system, offers the most accurate data with the least amount of *drift* and noise compared with the Xsens MVN suit. However, we have also shown in Paper III that such a multicamera optical marker-based system suffers not only from marker drop-out but also from camera *occlusion noise*.

Real-time performance. The tracking latency and jitter performance of the two evaluated MoCap systems were presented in Papers II and III. Such properties are important for real-time applications. Low latency is not the only important factor for control intimacy; jitter, a distortion of the time domain, can be equally important if high temporal precision is needed [65]. Additionally, it is important that the chosen MoCap system is able to deliver good quality and consistent MoCap data in real time. Extra filtering is necessary if the data is too noisy. Such filters will add latency and processing costs, as described in Chapter 4. The used network technology, which is responsible for delivering the MoCap data to the end application, is an additional real-time performance concern since it can contribute with latency, jitter, and even data frame dropouts (e.g. due to data loss in a wireless link).

System usability. It is important to consider the *usability* and what I have called the "out of lab" performance. Though the system works perfectly in the lab or in a specific environment, it may show surprisingly bad performance when used in a different environment or for a different task. This was found to be one of the biggest differences between the two MoCap systems compared.

In the experiment in Paper VII and for the Dance Jockey project, I chose to use two different MoCap systems. In both cases, the technology choices were based on how the MoCap technology fit the intended task. Good *data quality* was my main priority for the frequency experiment in Paper VII. During an experiment in a lab, it is possible to have some control of the environment. We could therefore minimize marker occlusion problems by carefully choosing a camera setup that fit the experiment. During this experiment we could also abandon recordings with corrupted data, e.g. marker drop outs. These factors made it reasonable to choose the OptiTrack system for this experiment.

For the Dance Jockey project, the Xsens suit was considered to be the most suitable system. Controlling the tracking environment is problematic when using a MoCap system for several performances on different distinct locations. We, therefore, needed a more environmental robust MoCap system. The real-time performance was also of high priority since this was a real-time application. Even though the OptiTrack system had some lower latency and jitter performance, the real-time data from the Xsens system was more consistent and robust. Additionally, the usability of the Xsens MVN system fit the task better. It was both easier to transport and set up. Finally, the *data quality* was found to be good enough for the intended tasks.

6.1.2 Developing the Dance Jockey system

Using full-body MoCap data for controlling sonic and musical features has shown to involve several challenges. The work was often experienced as frustrating since so many steps with experimenting and development were necessary to arrive at satisfactory performance levels. At the same time, it has also shown to offer many possibilities, and the system presented has given several hands-on solutions for how full-body MoCap data can be used to control sonic and musical features. Our most important discoveries are listed in the following.

Transition between states. Using full-body MoCap data for musical interaction was of special interest since it provided possibilities for using the body as a whole as the basis for musi-

cal interaction. Since our motivation was to build strong visible couplings between action and sound, we wanted to make a full-length performance piece in which all aspects of the performance were controlled solely by the Xsens MVN suit. In order to achieve this and, at the same time, also offer some varied content, we implemented a finite-state machine in the system. In this way, the performer could navigate between states that contained different action-sound mappings. The transitions and states were also used as active components of the performances, i.e. composition.

Ecological knowledge. Using full-body MoCap data offers many possibilities for controlling sonic and musical features. However, it can be difficult to determine how the MoCap data can be used to make good couplings between sound and motion. Here, we found it useful to consider our perceptual and cognitive constraints and our *ecological knowledge* of sound, meaning accumulated knowledge of sound and sound making and how they are related to the physical world. Taking inspiration from such ideas and the listed concepts in Section 2.5 was found fruitful since it guided our mappings to become more intuitive and easy to explore. We also found such mappings to give the most interesting couplings between motion and sound. This strategy became the main motivation behind most of the action-sound mappings we developed during the Dance Jockey project. We believe the audience could also gain from this strategy since such intuitive mappings should have an additional communicative value.

The gap of execution. Developing the Dance Jockey system demanded much work. Not only did it involve many mathematical and computational details, but there were also many possibilities to explore. When we wanted to try out an idea, it took days with development before we were able to try it out. Efficient tools are essential when attempting to compose and practice performances that employ full-body MoCap technology. Through developing our own tools and software while working with performance-related and technical aspects of the system, we were able to decrease the so-called *gap of execution*, or the gap between an idea and its realization. Such tools and software are, in my opinion, important for the creativity and spontaneity during composing and practicing performance pieces with full-body MoCap technologies.

6.1.3 Filtering real-time MoCap data

Processing MoCap data is essentially digital signal processing, and the most common processing approach in the time domain is *filtering*. As we have seen in this thesis, it is often necessary to process MoCap data in different ways before we can use them. Filtering real-time MoCap data has therefore been an important subject for this thesis, and the suggested best practices for filtering MoCap data for real-time applications are an important part of the contribution of this thesis. The work has consisted of developing tools, method and a range of filters applicable for real-time MoCap data. The following points summarize my findings:

MoCap data in the frequency domain. When wanting to filter MoCap data, it seems reasonable to regard MoCap data in the frequency domain, given the convincing results from the frequency experiment presented in Paper VII. The filter design methods based on the time

domain, such as Savitzky-Golay and other *polynomial fit* based approaches, do not offer the same customizability as the filter design methods based on the frequency domain. Polynomial fit approaches are only adjustable by the given polynomial order and the given filter lengths (see Section 4.7.2). It is also known that human motion does not necessarily follow polynomial curves [42]. In other words, I recommend applying filters that are designed and evaluated in the frequency domain since it seems to be the most effective approach for filtering MoCap data. Additionally, there are a large number of available DSP tools that function by specifying the desired frequency response.

Frequency analysis of MoCap data. To be able to design good application specific filters, it is necessary to determine the frequency content of the data that needs to be filtered. This was the goal of the experiment we presented in Paper VII. More specifically, we wanted to determine the typical frequency properties of free hand motion. To be able to analyze the collected MoCap recordings from the experiment, a general form of the *residual analysis* was developed (presented in Section 5.3.2). This method was found to be the most intuitive and robust for analyzing the frequency properties of recorded MoCap data. Based on this experiment, we have in Paper VII proposed to use a cutoff frequency between 5 and 15 Hz when filtering free hand motion.

Symmetric FIR filters: *The least square method.* *Symmetric FIR* filters are a sensible choice for post-processing of MoCap data, given their constant group delay error, i.e. *linear phase*, and the good availability of design methods. Since MoCap data can be considered to contain so-called *white noise*, I recommend using the least square method, e.g. the *firls* method in MATLAB, since it gives optimal symmetric FIR filters for such noise problems. However, if the filters are intended for real-time applications, the delay properties of the used filter become important. Unfortunately, symmetric FIR filters are not optimal if the lowest filter delay is wanted.

Proposed alternative filter design approach. As presented in Papers V and VI, the established filter design methods were found inadequate to design the wanted low-delay filters and low-pass differentiators. I have therefore in this thesis approached filter design with a heuristic method to be able to explore novel designs. Instead of trying to solve the nonlinear problem of IIR filter design analytically, I have used an alternative approach based on having a computer algorithm freely explore filter design following some heuristics. Defining the filter design problems as a multi-objective optimization problem has enabled me to consider trade-offs between conflicting objectives, which is a prominent challenge in filter design. And indeed, the proposed heuristic method found more optimal filters than the currently established methods can produce, as shown in Papers V, VI and Section 4.6 of this thesis.

Optimal low-delay filters. In this thesis, I have addressed the challenge of designing optimal digital filters with *low delay*, since I was unable to find research that targeted such challenges directly. The presented results in Paper V show that *unrestricted IIR (UR IIR) filters*, designed with the above proposed filter design method, offer the best combination of low delay and high noise attenuation. Such filters should be applicable for a wide range of real-time applications, e.g. computer games that use MoCap controllers. According to my

results in Section 4.7.3 and Paper V, asymmetric FIR filters can offer similar low-delay performance, although with higher filter order and greater computational cost. Given the experimental results in Paper V, it is also suggested that noise attenuation is linearly related with the filter delay. In other words, it is not possible to design filters with very low delay and large noise attenuation. However, by using the developed alternative design approach, we can custom-design filters with the wanted trade-off between the different filter design properties. In this way, it is possible to design the best possible filter for the given application.

Optimal low-pass differentiators. Most of the available MoCap system offers only spatial, i.e. positional and orientational, motion estimations. If properties like velocity or acceleration are wanted, it is necessary to use differentiators to compute the derivative of the spatial data. As we have shown in Paper III, MoCap systems are known to have different problematic noise properties. Though the noise only consists of submillimeter spikes, it gets heavily amplified in the differentiator process since the differentiator acts as a high-pass filter (see Section 4.2.3). Given this effect, I recommended to use so-called low-pass differentiators since they avoid the undesirable amplification of noise in the higher-frequency band. However, there are no established methods that offer such customizable design of IIR low-pass differentiators. In this thesis, I have therefore used the proposed alternative design method to design UR IIR low-pass differentiators. As shown in Paper VI, the presented UR IIR low-pass differentiators are shown to compare favorably with existing designs in literature.
Additionally, I have presented novel designs of *IIR low-pass differentiators of degree 2* with reduced delay in Paper VII. Using such differentiators is more optimal than using two low-pass differentiators of degree 1 in cascade. To my knowledge, such designs have not been presented before in the literature.

A range of proposed low-delay filters. Finally, in Paper VII, based on the above methods and results, we have presented a range of low-delay filters, including *low-pass filters* and *low-pass differentiators of degrees 1* and 2, which, in my experience, are the most useful filters for our target application. All filters have a group delay of 2 samples or less and have better low-delay performance than what currently established filter design methods can create. Compared with optimal symmetric FIR filters, they give a noise attenuation increase between 5 and 16 dB with similar delay or up to two to four times the delay reduction for similar magnitude properties. The proposed low-pass differentiators were especially found to offer a favorable combination of low passband error, high noise suppression and low group delay error. The specifications of these filters are given in the appendix of this thesis. Additionally, the proposed IIR filters are embedded in a MAX patch to provide easy access for non-engineers. The patch is presented in Section 5.3.2.

6.2 Conclusion

This PhD project has been concerned with the development of methods and technologies for using body motions for real-time musical interaction. This has included the evaluation of MoCap technologies, the development of the Dance Jockey system, and finally, the study of best practices for filtering MoCap data for real-time applications. The following points conclude the research:

- There are an increasing number of available MoCap technologies and two of the available systems have been evaluated in this thesis. According to the results, it is shown that both technologies provide their strengths and weaknesses. Since different applications will have different MoCap performance requirements, it is important to identify the needed performance criteria to be able to choose the best MoCap technology for the given task. The system known as Xsens MVN suit was found to be the most suitable system for real-time musical performances. This system was used for the Dance Jockey project.

- The development of the Dance Jockey system has shown many possibilities. However, many challenges were also encountered in the cumbersome course of using full body MoCap data for controlling sound and musical features. We have striven to achieve intuitive control concepts and tried to create a good match between action and sound through inspiration of our *ecological knowledge* of sound. Given the restricted time and resources, the Dance Jockey project has only been able to touch on the surfaces of the possibilities. However, the presented Dance Jockey system has given several hands-on solutions for how full-body MoCap data can be used to control sonic and musical features. The so-called *gap of execution*, or the gap between an idea and its realization, was identified as one of the biggest challenges during the creative process of composing and developing the performance pieces.

- To study best practices for filtering MoCap data for real-time applications, several methods and tools have been developed during the work of this thesis. First of all, the developed alternative filter design method has made it possible to design more optimal *low-delay noise filters* and more optimal *low-pass differentiators* than currently available. To be able to design application-specific filters, it was necessary to establish the frequency content of the MoCap data that we wanted to filter. In order to study this, we conducted an experiment and developed a tool to determine the generic frequency properties of free hand motion. Finally, based on the above methods and results, we have proposed a range of novel filters applicable for real-time musical interaction with MoCap systems. These filters are more optimal than the currently established design methods can produce. The filters are also applicable for other real-time applications that need the best possible filters with the lowest delay, e.g. computer games using MoCap controllers.

It can be concluded that this thesis has gathered knowledge about MoCap technologies, developed and demonstrated musical interaction with a full body MoCap, and studied and suggested best practices for filtering of MoCap data for real-time applications.

6.3 Future work

Given the broad goal that is targeted in this thesis, many challenges remain. First of all, it is necessary to conduct more quantitative evaluations of established and emerging MoCap technologies, e.g. new systems like *Leap Motion* and the *Xbox One Kinect*. Such evaluations are important to understand more of the strengths and weaknesses of the available system and how they can be used in musical applications. Affordable systems are of special interest since they are available for a larger community.

The Dance Jockey project has only touched on the many possibilities of how full body MoCap technologies can be used for musical interaction. What I see as the most prominent challenge is the so-called *gap of execution*. In this respect, I think it would be effective to establish a user-friendly *real-time MoCap toolbox*, which should consist of a range of powerful tools and methods for the effective *processing* of full-body MoCap data in real-time. The filters and filtering tools proposed in this thesis should be applicable for such a toolbox.

Extracting motion features from MoCap data is essentially *digital signal processing* (DSP), and according to my results, it is reasonable to regard MoCap data in the frequency domain. When wanting to process MoCap data, we can therefore use the waste number of already available and effective frequency-based DSP tools. In this respect, it would be interesting to investigate how the established DSP techniques could be applicable for extracting motion features, e.g. can band-pass filters and filter banks be of interest for us?

I have not been able to thoroughly test the range of filters presented in this thesis, and it is still necessary to understand more of the importance of the different filter features. For instance, it is possible to get higher noise attenuation by relaxing the group delay objective in the upper part of the passband. If the consequences of the different filter features are better understood, we can design filters that optimize the actual wanted filter performance. Finally, to make *unrestricted IIR* filter design with arbitrary specification readily available for designers, it is necessary to make a more computationally effective and user-friendly version of the filter design method I have proposed in this thesis.

Bibliography

[1] E. Altenmüller, J. Kesselring, and M. Wiesendanger. *Music, motor control and the brain*. Music, Motor Control and the Brain. Oxford University Press, 2006.

[2] R. K. Andy Hunt, Marcelo M. Wanderley. Towards a model for instrumental mapping in expert musical interaction. 2000.

[3] C. Bahn, T. Hahn, and D. Trueman. Physicality and feedback: a focus on the body in the performance of electronic music. In *Proceedings of the International Computer Music Conference*, 2001.

[4] R. Balakrishnan, T. Baudel, G. Kurtenbach, and G. Fitzmaurice. The rockin'mouse: integral 3d manipulation on a plane. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 311–318, New York, NY, USA, 1997. ACM.

[5] B. Caramiaux. *Studies on the Relationship between Gesture and Sound in Musical Performance*. PhD thesis, Universite Paris VI, Ircam Centre Pompidou, 2012.

[6] M. L. Chanda and D. J. Levitin. The neurochemistry of music. *Trends in Cognitive Sciences*, 17(4):179 – 193, 2013.

[7] A. Chottera and G. Jullien. A linear programming approach to recursive digital filter design with linear phase. *Circuits and Systems, IEEE Transactions on*, 29(3):139 – 149, mar 1982.

[8] E. F. Clarke. *Ways of Listening : An Ecological Approach to the Perception of Musical Meaning: An Ecological Approach to the Perception of Musical Meaning*. Oxford University Press, USA, 2005.

[9] P. Cook. Principles for designing computer music controllers. In *Proceedings of the 2001 conference on New interfaces for musical expression*, NIME '01, pages 1–4, Singapore, Singapore, 2001. National University of Singapore.

[10] G. Cortelazzo and M. Lightner. Simultaneous design in both magnitude and group-delay of IIR and FIR filters based on multiple criterion optimization. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(5):949 – 967, oct 1984.

[11] S. Dahl and A. Friberg. Visual perception of expressiveness in musicians body movements. *Music Perception*, 24:433–454, 2007.

[12] K. Deb. Multi-objective optimization. In E. K. Burke and G. Kendall, editors, *Search Methodologies*. Springer US, 2005.

[13] G. Essl and S. O'modhrain. An enactive approach to the design of new tangible musical instruments. *Org. Sound*, 11(3):285–296, 2006.

[14] W. W. Gaver. How do we hear in the world? explorations in ecological acoustics. *Ecological psychology*, 5(4):285–313, 1993.

[15] G. Giakas and V. Baltzopoulos. Optimal digital filtering requires a different cut-off frequency strategy for the determination of the higher derivatives. *Journal of biomechanics*, 30(8):851–855, 1997.

[16] J. Gibson. *The ecological approach to visual perception:*. Houghton Mifflin, 1979.

[17] R. Godøy and M. Leman. *Musical Gestures: Sound, Movement, and Meaning*. Taylor & Francis, 2009.

[18] R. I. Godøy. Motor-Mimetic Music Cognition. *Leonardo*, 36(4):317–319, Aug. 2003.

[19] R. I. Godøy. *Lecture Notes in Computer Science*, chapter Gestural Imagery in the Service of Musical Imagery, pages 99–100. Springer Berlin / Heidelberg, 2004.

[20] S. B. Gokturk, H. Yalcin, and C. Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 35–35. IEEE, 2004.

[21] K. Hinckley and R. J. K. Jacob. *Input/output Devices And Interaction Techniques*. CRC Press LLC: Boca, 2003.

[22] K. Holtzblatt. What makes things cool?: Intentional design for innovation. *Interactions*, 18(6):40–47, Novemeber 2011.

[23] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32:429–440, 2003.

[24] R. J. K. Jacob. New human-computer interaction techniques. Technical report, In BrouwerJanse, M., & Harrington, T. (Eds.), Human-Machine Communication for Educational Systems Design, 1994.

[25] R. J. K. Jacob. *The Computer Science and Engineering Handbook*, chapter Input Devices And Techniques, pages 1494–1511. CRC Press, 1996.

[26] R. J. K. Jacob, L. E. Sibert, D. C. McFarlane, and M. P. Mullen, Jr. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.*, 1(1):3–26, 1994.

[27] A. R. Jensenius. *ACTION - SOUND, Developing Methods and Tools to Study Music-Related Body Movement*. PhD thesis, University of Oslo, 2007.

[28] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman. Musical gestures: concepts and methods in research. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*, pages 12–35. Routledge, New York, 2010.

[29] S. Jordà. Digital instruments and players: Part i - efficiency and apprenticeship. In *Proceedings of the 2004 International Conference on New Interfaces for Musical Expression*, pages 59–63, 2004.

[30] X. Lai. Optimal Design of Nonlinear-Phase FIR Filters With Prescribed Phase Error. *Signal Processing, IEEE Trans.*, (9), sept. 2009.

[31] M. Lang. *Algorithms for the Constrained Design of Digital Filters with Arbitrary Magnitude and Phase Responses*. PhD thesis, The Vienna University of Technology, 1999.

[32] M. Lang. Least-squares design of IIR filters with prescribed magnitude and phase responses and a pole radius constraint. *Signal Processing, IEEE Transactions on*, 48(11):3109 –3121, nov 2000.

[33] D. J. Levitin, S. McAdams, and R. L. Adams. Control parameters for musical instruments: a foundation for new mappings of gesture to sound. *Org. Sound*, 7(2):171–189, 2002.

[34] W.-S. Lu, S.-C. Pei, and C.-C. Tseng. A weighted least-squares method for the design of stable 1-d and 2-d iir digital filters. *Signal Processing, IEEE Transactions on*, 46(1):1 –10, jan 1998.

[35] M. T. Marshall and M. M. Wanderley. Vibrotactile feedback in digital musical instruments. In *NIME 06: Proceedings of the 2006 conference on New interfaces for musical expression*, pages 226–229, Paris, France, France, 2006. IRCAM — Centre Pompidou.

[36] E. R. Miranda and M. Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard (Computer Music and Digital Audio Series)*. A-R Editions, Inc., 1st edition.

[37] S. Mitra. *Digital signal processing: a computer based approach*. McGraw-Hill H. E., 2005.

[38] A. G. E. Mulder. *Design of Virtual Three-dimensional Instruments for Sound Control*. PhD thesis, Simon Fraser University, 1998.

[39] M. Nusseck and M. M. Wanderley. Music and motion - how music related ancillary body movements contribute to the experience of music. *Music Perception*, 26:335–353, 2008.

[40] K. Nymoen. *Methods and Technologies for Analysing Links Between Musical Sound and Body Motion*. PhD thesis, University of Oslo, 2013.

[41] T. W. Parks and C. Burrus. *Digital filter design*. Topics in digital signal processing. Wiley, 1987.

[42] D. Robertson. *Research Methods in Biomechanics*. Human Kinetics, 2004.

[43] D. Rosenberg, H. Luinge, and P. Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Technologies*, 2009.

[44] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (3rd Edition)*. Prentice Hall, 3 edition, 2009.

[45] J. Ryan. Some remarks on musical instrument design at steim. *Contemporary Music Review*, 6(1):3–17, 1991.

[46] T. Saramäki and J. Yli-kaakinen. Design of digital filters and filter banks by optimization: Applications. In *EUPSICO 2000: European signal processing conference*, 2000.

[47] R. Schafer. What is a savitzky-golay filter? *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

[48] A. Schmeder, A. Freed, and D. Wessel. Best practices for open sound control. In *Linux Audio Conference*, Utrecht, NL, 01/05/2010 2010.

[49] I. Selesnick. Maximally flat low-pass digital differentiator. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 49(3):219 –223, mar 2002.

[50] I. W. Selesnick and C. S. Burrus. Maximally Flat Lowpass FIR Filters with Reduced Delay. *IEEE TRANS. ON CIRCUITS AND SYSTEMS II*, 45:53–68, 1998.

[51] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2007.

[52] D. Simon. Kalman filtering. *Embedded Systems Programming*, 14(6):72–79, 2001.

[53] S. A. Skogstad, S. Holm, and M. Hovin. Digital IIR Filters With Minimal Group Delay for Real-Time Applications. In *Engineering and Technology (ICET), 2012 International Conference on*. ICET, IEEE Computer Society, 2012.

[54] S. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Tech. Pub., 1997.

[55] R. Storn. Differential evolution design of an IIR-filter. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 268 –273, may 1996.

[56] M. Thompson. *The Application of Motion Capture to Embodied Music Cognition Research*. PhD thesis, UNIVERSITY OF JYVÄSKYLÄ, 2012.

[57] C.-J. Tsay. Sight over sound in the judgment of music performance. *Proceedings of the National Academy of Sciences*, Aug. 2013.

[58] R. Vertegaal, T. Ungvary, and M. Kieslinger. Towards a musician's cockpit: Transducers, feedback and musical function. In *Proceedings of the 1996 International Computer Music Conference*, pages 308–311. The International Computer Music Association, 1996.

[59] M. M. Wanderley. Gestural control of music. In *Proceedings of the International Workshop on Human Supervision and Control in Engineering and Music.*, pages 101–130, 2001.

[60] M. M. Wanderley and N. Orio. Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Comput. Music J.*, 26(3):62–76, Sept. 2002.

[61] Y. Wang, B. Li, and Y. Chen. Digital IIR filter design using multi-objective optimization evolutionary algorithm. *Applied Soft Computing*, 11(2):1851 – 1857, 2011.

[62] R. Wechsler. Applications of motion tracking in making music for persons with disabilities. In *Proceedings of the 4. Workshop of» Innovative Computerbasierte Musikinterfaces «, Mensch&Computer, Konstanz, Oldenbourg Wissenschaftsverlag*, 2012.

[63] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.

[64] G. Welch and E. Foxlin. Motion tracking: no silver bullet, but a respectable arsenal. *Computer Graphics and Applications, IEEE*, 22(6):24 –38, nov.-dec. 2002.

[65] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, Sept. 2002.

[66] J. S. Wilson. *Sensor technology handbook*, volume 1. Newnes, 2005.

[67] S. Winder. *Analog and digital filter design*. EDN series for design engineers. Newnes, 2002.

[68] D. Winter. *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons, 2009.

[69] G. Wood. Data smoothing and differentiation procedures in biomechanics. *Exercise and sport sciences reviews*, 10(1):308, 1982.

[70] M. Zentner and T. Eerola. Rhythmic engagement with music in infancy. *Proceedings of the National Academy of Sciences of the United States of America*, 107(13):5768–5773, march 2010.

[71] Z. Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10, 2012.

Papers

I Using IR Optical Marker Based Motion Capture for Exploring Musical Interaction.
S.A. Skogstad, A.R. Jensenius and K. Nymoen
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 407–410 Sydney University of Technology 2010.

II OSC Implementation and Evaluation of the Xsens MVN suit.
S.A. Skogstad, K. Nymoen, Y.d. Quay and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 300–303, University of Oslo 2011.

III Comparing Inertial and Optical MoCap Technologies for Synthesis Control.
S.A. Skogstad, K. Nymoen, and M.E. Høvin.
In *Proceedings of SMC 2011 8th Sound and Music Computing Conference "Creativity rethinks science"*, pages 421–426, Padova University Press 2011.

IV Developing the Dance Jockey System for Musical Interaction with the Xsens MVN suit.
S.A. Skogstad, K. Nymoen, Y.d. Quay and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 226–229, University of Michigan 2012.

V Digital IIR Filters With Minimal Group Delay for Real-Time Applications.
S.A. Skogstad, S. Holm and M.E. Høvin.
In *IEEE The International Conference on Engineering and Technology. 2012.*, pages 1–6, German University in Cairo 2012.

VI Designing Digital IIR Low-Pass Differentiators With Multi-Objective Optimization.
S.A. Skogstad, S. Holm and M.E. Høvin.
In *IEEE 11th International Conference on Signal Processing. 2012.*, pages 10–15, Beijing Jiaotong University 2012.

VII Filtering Motion Capture Data for Real-Time Applications.
S.A. Skogstad, K. Nymoen, S. Holm, M.E. Høvin and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 196–197, Kaist University, Daejeon 2013.

Paper I

Using IR Optical Marker Based Motion Capture for Exploring Musical Interaction.
S.A. Skogstad, A.R. Jensenius and K. Nymoen
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 407–410, Sydney University of Technology 2010.

Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010), Sydney, Australia

Using IR Optical Marker Based Motion Capture for Exploring Musical Interaction

Ståle A. Skogstad,[♮] Alexander R. Jensenius,[♯] Kristian Nymoen[♮]

[♮]University of Oslo
Department of Informatics
Pb 1080 Blindern, 0316 Oslo, Norway
{savskogs, krisny}@ifi.uio.no

[♯]University of Oslo
Department of Musicology
Pb 1017, Blindern, 0315 Oslo, Norway
a.r.jensenius@imv.uio.no

ABSTRACT

The paper presents a conceptual overview of how optical infrared marker based motion capture systems (IrMoCap) can be used in musical interaction. First we present a review of related work of using IrMoCap for musical control. This is followed by a discussion of possible features which can be exploited. Finally, the question of mapping movement features to sound features is presented and discussed.

1. INTRODUCTION

Motion capture (MoCap) is a term often used to describe the process of recording human body movement and storing it in the digital domain. Many different disciplines make use of MoCap systems, and they can briefly be divided into two groups: *analysis* and *synthesis*. The first approach (analysis) is typically found in fields working on bio-mechanical research questions, e.g. medicine, rehabilitation and sports science. The second approach (synthesis) can be found in the entertainment sector, where MoCap systems are used to create lifelike animations in movies and computer games.

Many different MoCap technologies exist [1], and we will here choose to split them into two different groups: *optical* and *non-optical* systems. Among the non-optical systems, one of the most affordable solutions is that of *inertial* sensor systems, based on sensors such as gyroscopes, accelerometers and magnetometers. While each such sensor outputs relevant movement data in themselves, MoCap systems based on such sensors typically perform *sensor fusion* on the raw data. Sensor fusion means that data from the individual sensors are combined such that it is possible to integrate the data to calculate position (and sometimes orientation) with fairly little drift. On the positive side, such systems are often portable and flexible, and provide good value for money. Unfortunately, they often provide poorer spatial accuracy and precision than optical systems, and have problems with the measured position drifting over time.

Mechanical MoCap systems are based on directly tracking the angles of body joints through the use of flex sensors. Such systems are often flexible and durable, and have been used for many creative applications.

Magnetic systems calculate both 3D position and 3D orientation based on moving a coil in an electromagnetic field. They often give precise and reliable data, but have a comparably small capture volume. Another big drawback is the susceptibility to magnetic and electrical interference.

While they have many positive sides, inertial, mechanical and magnetic systems share one problem: they usually rely on fairly large sensors that have to be attached with cables to the computer. Exactly this is what makes *optical* MoCap systems preferable in many contexts, since they provide for a non-obtrusive and flexible solution.

Optical systems can be divided into visual *markerless* systems and *marker based* systems. Both these techniques rely on computer vision techniques for extracting movement features and tracking body parts. Although markerless computer vision techniques are in rapid development, the marker based solutions still make for more accurate, precise and fast tracking. Optical MoCap has been particularly popular for creative applications, due to the low cost, flexibility and availability of relevant tools, e.g. Max/MSP/Jitter and EyesWeb [3].

The technique which is often referred to as state of the art in the world of MoCap, is what could be called *optical infrared marker based motion capture* (IrMoCap). This is based on a group of cameras, typically no less than 6, surrounding the person(s)/object(s) to be tracked. The cameras emit infrared light which is bounced off reflective markers attached on the body of the person being observed and captured by the cameras. Through triangulation techniques the system calculates the absolute position in space, with submillimeter resolution and at speeds above 500 Hz. By combining multiple markers it is possible to uniquely identify certain objects, something which may also be accomplished using active markers that emit their own light.

We have experience with all of the above mentioned MoCap solutions, and see that they all have positive and negative effects. In our current research, however, we have decided to focus our attention on IrMoCap, since this is the technique which currently provides for the most precise, accurate and fast MoCap solution. On the negative side they are expensive and requires a controlled lab setting to work properly. This is because the system needs to be calibrated thoroughly and is sensitive to light pollution. Despite these drawbacks, we believe that the knowledge and experience gained from using such systems may be transferred to other more accessible and affordable MoCap technologies in the future.

Our main research goal is to explore the control potential of human body movement in musical applications. By combining high quality MoCap data with advanced machine learning techniques, we try to explore multidimensional mappings between motion features and sound features. Here we are interested in exploring everything from

direct control, like playing an instrument, to more *indirect* control, i.e. controlling more global features in the sound and musical structures. We want, in other words, to explore the possibilities of using new technologies to increase the connection between human motion and musical expression.

2. RELATED WORK

We have only found a few studies that have been published on using IrMoCap systems in musical interaction, and we have chosen to separate this into two categories: non-real-time and real-time.

2.1 Non-real-time control

Dobrian et al. describes a system where data recorded with an IrMoCap system can be mapped to MIDI signals [5]. Their software makes it possible to choose which marker and its associated motion feature that should be mapped. The motion features include marker position, velocity, acceleration, and distance and between markers (in one, two or three dimensions). In addition to linear mappings, the software also allows for reversed, exponential, logarithmic mappings.

An important point that Dobrian et al. reflects upon, is that performing on a 'touchless' instrument both provides a challenge, but also opens for interesting musical explorations. We also share their interest in trying to develop strategies for keeping multidimensionality (e.g. data from 30 3D markers) throughout the mapping process.

One of the challenges when working with IrMoCap is the massive amounts of data that has be to handled, e.g. 30x3 marker values for each recorded frame. Bevilacqua et al. report on developing techniques for segmentation of the movement stream and what they call 'gestural segmentation' in [2]. Here they describe some of the numerical problems of computing velocity and acceleration from noisy data and point out that filtering is important, but that it also adds latency to the system. They experimented with using principal component analysis (PCA) for feature extraction, and using the output for controlling MIDI systems and signal processing.

2.2 Real-time interaction

The first example we have found of using IrMoCap in real-time musical applications is a project by Qian et al., in which they used "a number of static human body gestures (poses) to drive the interactive system" [21]. They divided the body into 10 rigid 'objects,' and used angular relations as features for the pattern recognition classification. This was used to control granular and additive sound synthesis, where pitch material were selected through a simple genetic algorithm. Unfortunately, we have not been able to find any video examples of their performance to evaluate the approach.

Other examples of real-time applications include Woolford's use of IrMoCap to visualize and sonify body motion in installations [25], and Downie's experimentation in a stage setting [7]. We see that many research groups get access to and set up projects around IrMoCap technologies, one example being the *Embodied Generative Music* project at IEM in Graz [9]. They have been experimenting with an installation where you prerecorded music is 'laid out' in physical space, and where it is possible to explore the "tactile" feeling of sound in space.

2.3 Sonification

A related but still different approach is that of Kapur et al., where the goal is to build the necessary infrastructure to study the use of sonification for understanding human motion [17]. They are interested in studying how the musician's posture and movement during performance affect the sound produced, as well as the emotional content of the performance. They also hope that studying sonification of IrMoCap data can aid individuals with motor disorders. The study did not involve real-time examples but used recorded data of people performing music (tabla and violin), dancers acting out different emotions, and individuals having impairments in sensory motor coordination. The sonifications consist of mapping marker positions to control sinusoidal oscillators, FM synthesis, phase vocoders and physical models of instruments.

In the same direction we find work related to sonification of IrMoCap data from musicians' 'ancillary gestures', with the aim of providing an alternative perspective when analyzing movements of musicians [23, 11]. This was also done by Larkin et al. in a project where IrMoCap data of string performers were sonified, intended as an interactive feedback to the performer [18]. Vogt et al. have a similar approach with applications in physiotherapy and other training contexts [24].

3. MOTION EXTRACTION

Our research goal is to study the capabilities of IrMoCap in the context of musical expression. The challenge then is to develop solutions for extracting meaningful information from the continuous stream of data, and map these to relevant features in the musical sound. This is both a question about making an interpretation of the data, but also a technical challenge when it comes to handling marker occlusion problems, data noise, latency and computational and numerical challenges.

In the context of optical MoCap, Camurri et al. [4] have suggested a four-layer framework that can be useful for our application:

- Layer 1: Physical signals
- Layer 2: Low-level features
- Layer 3: Mid-level features
- Layer 4: Concepts and structures

Separating between the different layers may help to structure some of the challenges, both conceptual and technological, and will form the basis for our thinking about IrMoCap data processing in the following sections.

3.1 Marker and Object Data

The first and second layers in the model of Camurri are related to the physical signals and low-level features, and is related to the output we get from a IrMoCap system: 3D positions of the markers that the cameras can see. These markers, passive or active, can be placed directly on the human body or placed on objects that can be moved in the space.

In addition to tracking the position of an object, it is also possible to find the angular orientation of an object by placing 3 or more markers on the object's surface. Here we are experimenting with having many objects, all with unique marker constellations, so that it is possible to uniquely identify all the objects. This will make it possible to play with all these objects in the motion capture area simultaneously.

3.2 Mapping Markers to a Kinematic Model

Instead of dealing with a vast amount of isolated markers and/or 6D objects, we are also exploring techniques for grouping them together and study how they move in relation to each other. This can be accomplished by defining one or more *kinematic models*, e.g. of the human body. But it can also be possible to define kinematic models for

Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010), Sydney, Australia

other types of composite systems, e.g. a movable sculpture. Defining a kinematic model can be done by representing the data as several connected solid objects with the respected joint angles between adjacent solid body parts [21]. A benefit of such an approach is that it helps in decreasing the dimensions of the data set, and can provide us with more meaningful data.

3.3 Manipulation of Parameters

There are endless possibilities for manipulation of the above mentioned parameters: change the scale of axes, invert signals etc. It is also possible to extract different relationships between markers, e.g. relative distance and angles between points. Further on, it is possible to perform numerical calculation on the output streams to obtain properties like velocity, acceleration, jerk etc. All of these, however, are only numerical approximations, and noise from the data will propagate through the computations and possibly be amplified by the numerical algorithms [5]. These numerical computations should therefore be done with care. Filtering is a possible solution to get less noisy results, but a filter and other computations will at the same time add latency to the system.

3.4 Spatial aspects

Moving towards mid-level features, there are many questions when it comes to how to extract meaningful information from the continuous data sets. One approach here is to look at spatial aspects of the data. A kinematic model of the human body can be a good starting point for extracting information about specific body postures and placement of the body in space. Information about different body postures can for example be mapped to different sound features, and it may be possible to morph between discrete postures.

3.5 Temporal Aspects

Instead of (or addition to) the spatial aspects, we can work with temporal aspects. Placement of sonic objects in time is an underlying feature in the development of musical structures, so we need to find solutions for identifying, representing and utilizing temporal features from MoCap data. Here it can help to think about a three-level model of temporality: *sub-chunk*, *chunk* and *supra-chunk* [10]. Here the chunk level represents a time span of approximately 1-5 seconds, a time span which fits well with our working memory. The chunk level also (not coincidentally) happen to cover the time span of human actions, speech and music phrasing. In this model of time, the sub-chunk level is related to short sensations, while the supra-chunk level can be thought of as made up of a series of chunks. If we think about the continuous stream of MoCap data as the sub-chunk level, then segmentation of this stream into action segments that fall within the range of 1-5 seconds would correspond to the chunk level.

3.6 Pattern Recognition

As mentioned above, pattern recognition techniques have been used for mapping motion to sound [2, 21]. The typical goal here would be to recognize various types of expressive features from body movement and map these to relevant sounds. Here the dimensionality of the feature space is important for the robustness of recognition rates [8]. For example using 30 3D marker streams directly as features to the classifier can be problematic. This can be solved by reducing the dimensionality in the spatial and/or temporal domains, as mentioned above. Also, standard dimensionality reduction techniques from the field of pattern recognition can be used to find the features that work best.

An important conceptual question is how pattern recognition algorithms can support our goals. Using pattern recognition can certainly give us more options for the mapping to musical features, but how can it be used in an interesting way? We believe it is important that the final artistic results should be something new that we cannot do with traditional techniques. Simple one-to-one mappings, and trigger based systems would not do justice to the richness and complexity afforded by the IrMoCap system. The artistic result can end up just being a demonstration of technology with (hopefully) more than 90% correct recognition rate. An added challenge is that we are not good at reproducing our action precisely [19].

4. MAPPING MOTION TO SOUND

After evaluating some of the challenges when it comes to retrieving, processing and exploring data from an IrMoCap system in the previous section, we will here look at some of the challenges when it comes to mapping such data to sound features. This is a broad field and we will only touch on some of its complexity.

4.1 Sound-producing actions

Looking at the sound-producing actions used when performing a musical instrument, they can typically be divided into two groups: *excitation* and *modification* actions [15]. We can further distinguish between two types of excitations: *discrete* (e.g. triggers) or *continuous* excitation (e.g. bowing).

The raw data from an IrMoCap system is a continuous stream of numbers, so if we want to trigger signals we need to identify discrete actions through segmentation. The question, then, is whether using such a system for trigging predefined sounds is particularly interesting, or whether we might be better off by using an extra controller with simple buttons. This touches some of the challenges when it comes to designing connections between motion and musical features; to be effective the mapping should somehow match our mental model of what we want to control [22]. At the same time, several studies have shown that users find more complex and composite mappings more musically challenging and interesting [12, 16].

4.2 Touchless Actions

We can define *touchless action* as an action 'in the air' and where we cannot use the haptic and tactile response of a normal physical controller to guide us. In a musical context this implies a virtual relationship between sound and action since the relationship between the two is not bound by physical laws like we find in acoustic instruments [14].

When designing control interfaces for normal desktop computers, the design goals are rather straight forward. The interfaces should be ergonomic and effective, properties which are relatively easy to measure. Musical interfaces, on the other hand, have the extra requirement of being artistically interesting to use, a quality which is hard to evaluate and determine [16]. One design aspect which is especially important for virtual instruments is how the instrument's functionality can be understood mentally [22]. If the instrument is virtual, our whole comprehension of the instrument must either come from the sonic feedback or from our bodily experience of using the instrument. It seems plausible that the understanding of the connection between action and sound is a crucial point for the playability of a virtual instrument, but equally so for the audience watching the performance [6].

If we want to use touchless action as the basis for controlling musical features, it may be relevant to consider to

what degree we are conscious about our own body and its motion. If we use physical properties of tracked motion we need to take into account how these properties are understood by the users. For example so called *naive physics*, the untrained human perception of basic physical phenomena, can differ from what the data tells us [13]. Therefore, when using features like acceleration it is not certain that the user's understanding of these features reflects the numerical values.

A question connected to the potential of using touchless actions as control data is how many dimensions our actions consist of. Or maybe more important, how many dimensions are we able to exploit as control data? It may be appropriate to study the *informational theoretical* content. What is the needed sampling rate and how many bits per second are our touchless actions able to communicate?

Several groups of people are trained in touchless action. Dancers are experts in doing technically difficult actions, hearing impaired are experts in sign language and all of us use body language in our everyday life. To be able to exploit touchless action in a musical setting is certainly an interesting idea. But probably new paradigms are needed to map these actions to meaningful musical features. Until then it may be a good idea to design virtual instruments by mimicking aspects of our physical world so that we can take advantage of our established ecological experience of living in the world [19, 13].

4.3 Mapping to Sound Features

Let us briefly look at some possibilities when it comes to translating various types of motion and action features to sonic and musical features. A simple example is to map absolute marker position to the pitch of a sound. This may seem like a trivial task, but involves many different possibilities: should it be continuous control of pitch or in steps? How does pitch space relate to physical space? What types of pitch resolution and scales should be used? Instead of using absolute marker position to control sound features, it is also possible to look at the relative distance or angular position between two or more markers. These and many other similar questions will be the subject of some of our systematic studies of relationships between motion and sound in the coming years.

4.4 Spatialization

Another approach we are going to investigate in future studies include that of *spatialization*, i.e. placement of sound in space. The addition of a 32 channel speaker system in our motion capture lab provides the opportunity to explore control of sound through position and motion of the body in space. This may include moving sound sources around in the space, but also studying more complex relationships between physical and sonic space.

One approach to start such exploration may be to start by randomly setting up mappings between motion and sound features, much in the same way as the video to sound sonification suggested by Pelletier [20]. Instead of using optical flow we can let the marker displacement be sonified with additive or granular synthesis, something which may hopefully result in a rich combined motion and sound experience. Here marker occlusion and noise will also not be so problematic as long as a high percentage of the markers is properly tracked.

5. CONCLUSION

Infrared optical marker based motion capture technology is currently the state of art of motion capture systems, and despite some limitations, we believe such systems may provide for interesting and inspirational exploration of what other motion capture technologies can be used for. This paper has provided a review of some related work, and has covered some of the challenges related to using such systems in musical interaction. Much research still remains to make good musical use of such technologies. Here we believe it is reasonable to start by mimicking the already known physical world.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/motion_capture.
[2] F. Bevilacqua, J. Ridenou, and D. Cuccia. 3D motion capture data: motion analysis and mapping to music. In *SIMS*, 2002.
[3] A. Camurri. Toward real-time multimodal processing: Eyesweb 4.0. In *Proc. AISB*, 2004.
[4] A. Camurri et al. Multimodal analysis of expressive gesture in music and dance performances. *LNCS*, 2004.
[5] C. Dobrian and F. Bevilacqua. Gestural control of music: using the vicon 8 motion capture system. In *NIME*, 2003.
[6] C. Dobrian and D. Koppelman. The 'e' in nime: musical expression with new computer interfaces. In *NIME*, 2006.
[7] M. Downie. *Choreographing the Extended Agent: performance graphics for dance theater*. PhD thesis, Massachusetts Institute of Technology, 2005.
[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
[9] G. Eckel and D. Pirr. On artistic research in the context of the project embodied generative music. In *ICMC*, 2009.
[10] R. I. Godøy. *Systematic and Comparative Musicology: Concepts, Methods, Findings.*, chapter Reflections on chunking in music., pages 117–132. Peter Lang, 2008.
[11] F. Grond, T. Hermann, V. Verfaille, and M. Wanderley. Methods for effective ancillary gesture sonification of clarinetists. In *LNCS*, 2009.
[12] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. 2002.
[13] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum. Reality-based interaction: a framework for post-wimp interfaces. In *CHI*, pages 201–210, 2008.
[14] A. R. Jensenius. *ACTION - SOUND, Developing Methods and Tools to Study Music-Related Body Movement*. PhD thesis, University of Oslo, 2007.
[15] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman. Musical gestures: concepts and methods in research. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*, pages 12–35. Routledge, New York, 2010.
[16] S. Jordà. Digital instruments and players: Part i - efficiency and apprenticeship. *NIME*, 2004.
[17] A. Kapur and G. Tzanetakis. A framework for sonification of vicon motion capture data. In *Proc. DAFX05*, 2005.
[18] O. Larkin, T. Koerselman, B. Ong, and K. Ng. Sonification of bowing features for string instrument training. In *ICAD*, 2008.
[19] A. G. Mulder, S. S. Fels, and K. Mase. Design of virtual 3D instruments for musical interaction. In *GI*, 1999.
[20] J.-M. Pelletier. Sonified motion flow fields as a means of musical expression. In *Proc. NIME*, 2008.
[21] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis. A gesture-driven multimodal interactive dance system. In *ICME*, 2004.
[22] S. Skogstad. Models for the design of interfaces for musical expression. *To be submitted*.
[23] V. Verfaille, O. Quek, and M. M. Wanderley. Sonification of musicians' ancillary gestures. In *Proc. ICAD*, 2006.
[24] K. Vogt, D. Pirr, I. Kobenz, R. Hlldrich, and G. Eckel. Physiosonic - movement sonification as auditory feedback. Copenhagen, Denmark, 2009.
[25] K. Woolford. Will.0.w1sp - installation overview. In *ACM Multimedia (MM'07)*, 2007.

Paper II

OSC Implementation and Evaluation of the Xsens MVN suit.
S.A. Skogstad, K. Nymoen, Y.d. Quay and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 300–303, University of Oslo 2011.

Proceedings of the International Conference on New Interfaces for Musical Expression, 30 May - 1 June 2011, Oslo, Norway

OSC Implementation and Evaluation of the Xsens MVN suit

Ståle A. Skogstad and Kristian Nymoen
fourMs group - Music, Mind, Motion, Machines
University of Oslo, Department of Informatics
{savskogs,krisny}@ifi.uio.no

Yago de Quay
University of Porto, Faculty of Engineering
Rua Dr. Roberto Frias, s/n
4200-465 Portugal
yagodequay@gmail.com

Alexander Refsum Jensenius
fourMs group - Music, Mind, Motion, Machines
University of Oslo, Department of Musicology
a.r.jensenius@imv.uio.no

ABSTRACT

The paper presents research about implementing a full body inertial motion capture system, the Xsens MVN suit, for musical interaction. Three different approaches for streaming real time and prerecorded motion capture data with Open Sound Control have been implemented. Furthermore, we present technical performance details and our experience with the motion capture system in realistic practice.

1. INTRODUCTION

Motion Capture, or MoCap, is a term used to describe the process of recording movement and translating it to the digital domain. It is used in several disciplines, especially for bio-mechanical studies in sports and health and for making lifelike natural animations in movies and computer games. There exist several technologies for motion capture [1]. The most accurate and fastest technology is probably the so-called infra-red optical marker based motion capture systems (IrMoCap)[11].

Inertial MoCap systems are based on sensors like accelerometers, gyroscopes and magnetometers, and perform *sensor fusion* to combine their output data to produce a more drift free position and orientation estimation. In our latest research we have used a commercially available full body inertial MoCap system, the Xsens MVN[1] suit [9]. This system is characterized by having a quick setup time and being portable, wireless, moderately unobtrusive, and, in our experience, a relatively robust system for on-stage performances. IrMoCap systems on the other hand have a higher resolution in both time an space, but lack these stage-friendly properties. See [2] for a comparison of Xsens MVN and an IrMoCap system for clinical gait analysis.

Our main research goal is to explore the control potential of human body movement in musical applications. New MoCap technologies and advanced computer systems bring new possibilities of how to connect human actions with musical expressions. We want to explore these possibilities and see how we can increase the connection between the human body's motion and musical expression; not only focusing on the performer, but also on how the audience perceives the performance.

To our knowledge, we are among the first to use a *full body* inertial sensor based motion capture suit in a musical setting, and hence little related work exists. Lympouridis et. al. has used the inertial system Orient-2/-3 for sonification of *gestures* and created a framework for "bringing together dancers, composers and musicians" [6][5]. Meas et. al have used 5 inertial (Xsens) sensors to quantify the relation between sound stimuli and bodily response of subjects [7]. An upper body mechanical system has briefly been examined by [3]. See [11] for a review of related work in the area of IrMoCap for musical interaction.

In the next section, we will give a brief overview of the Xsens MVN technology. Then in section 3 we will report on three Open Sound Control implementations for the Xsens system and discuss some of our reflections. In section 4 we will give our evaluation and experience with the Xsens MVN system, before we propose a technology independent real time MoCap toolbox in section 5.

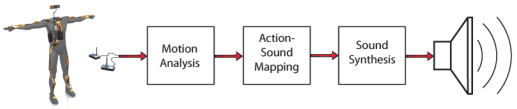


Figure 1: The Xsens suit and possible data flow when using it for musical interaction.

2. THE XSENS MVN TECHNOLOGY

The Xsens MVN technology can be divided into two parts. First, the sensor and communication hardware are responsible for collecting and transmitting the raw sensor data. Second, these data are treated by the Xsens MVN software engine, which interprets and reconstructs the data to full body motion while trying to minimize drift.

2.1 The Xsens MVN Suit (Hardware)

The Xsens MVN suit consists of 17 inertial MTx sensors, which are attached to key areas of the human body [9]. Each sensor consists of a 3D gyroscope, 3D accelerometer and magnetometer. The raw signals from the sensors are connected to a pair of Bluetooth 2.0 based wireless transmitters, which transmit the raw motion capture data to a pair of wireless receivers. The total weight of the suit is approximately 1.9 kg and the whole system comes in a suitcase with the total weight of 11 kg.

2.2 The Xsens MVN engine (Software)

The data from the Xsens MVN suit is fed to the MVN software engine that uses sensor fusion algorithms to produce

[1] *Xsens MVN* (MVN is a name not an abbreviation) is a motion capture system designed for the human body and is not a generic motion capture device.

Proceedings of the International Conference on New Interfaces for Musical Expression, 30 May - 1 June 2011, Oslo, Norway

absolute orientation values, which are used to transform the 3D linear accelerations to global coordinates. These in turn are translated to a human body model which implements joint constraints to minimize integration drift [9].

The Xsens MVN system outputs information about body motion by expressing body postures sampled at a rate up to 120Hz. The postures are modelled by 23 body segments interconnected with 22 joints [9]. The Xsens company offers two possibilities of using the MVN fusion engine: the Windows based *Xsens MVN Studio* and a software development kit called *Xsens MVN SDK*.

2.3 How to use the System

There are three main suit configurations; full body, upper body or lower body. When the suit is properly configured, calibration is needed to initialize the position and orientation of the different body segments. When we are satisfied with the calibration the system can be used to stream the motion data to other applications in real-time or perform recordings for later playback and analysis.

How precise one needs to perform the calibration may vary. We have found that so-called *N-pose* and *T-pose* calibrations are the most important. A *hand touch* calibration is recommended if a good relative position performance between the left and right hand is wanted. Recalibration can be necessary when the system is used over a longer period of time. It is also possible to input body measurements of the tracked subject to the MVN engine, but we have not investigated if this extra calibration step improves the quality of data for our use.

In our experience, setting up the system can easily be done in less than 15 minutes compared to several hours for IrMoCap systems [2].

2.4 Xsens MVN for Musical Interaction

A typical model for using the Xsens suit for musical application is shown in Figure 1. In most cases, motion data from the Xsens system must be processed before it can be used as control data for the sound engine. The complexity of this stage can vary from simple scaling of position data to more complex pattern recognition algorithms that look for mid/higher-level cues in the data. We will refer to this stage as *cooking* the motion capture data.

The main challenges of using the Xsens suit for musical interaction fall into two interconnected groups. Firstly, the purely technical challenges, such as minimizing latency, managing network protocols and handling data. Secondly, the more artistic challenges involving questions like how to make an aesthetically pleasing connection between action and sound. This paper will mainly cover the technical challenges.

3. IMPLEMENTATION

To be able to use the Xsens MVN system for musical interaction, we need a way to communicate the data that the system senses to our musical applications. It was natural to implement the OSC standard since the Xsens MVN system offers motion data which is not easily related to MIDI signals. OSC messages are also potentially easier to interpret since these can be written in a human readable form.

3.1 Latency and Architecture Consideration

Low and stable latency is an important concern for *real-time* musical control [12]. This is therefore an important issue to consider when designing our system. Unfortunately, running software and sending OSC messages over normal computer networks offers inadequate support for synchronization mechanisms, since standard operating systems do not support this without dedicated hardware [10]. In our experience, to get low latency from the Xsens system, the software needs to run on a fast computer that is not overloaded with other demanding tasks. But how can we further minimize the latency?

3.1.1 Distribution of the Computational Load

From Figure 1 we can identify three main computationally demanding tasks that the data need to traverse before ending up as sound. If these tasks are especially demanding, it may be beneficial to distribute these computational loads to different computers. In this way we can prevent a computer from suffering too much from computational load, which can lead to a dramatic increase of latency and jitter. This is possible with fast network links and a software architecture that supports the distribution of computational loads. However, it comes at the cost of extra network overhead, so one needs to check if the extra cost does not exceed the benefits.

3.1.2 The Needed Communication Bandwidth

The amount of data sent through a network will partly be related to the experienced network latency. For instance, we should try to keep the size of the OSC bundles lower than the maximum network buffer size,[2] if the lowest possible network latency is wanted. If not, the bundle will be divided into several packages [10]. To achieve this, it is necessary to restrict the amount of data sent. If a large variety of data is needed, we can create a dynamic system that turns different data streams on when needed.

3.2 OSC Implementations

There are two options for using the Xsens MVN motion data in real time, either we can use the Xsens Studio's UDP network stream, or make a dedicated application with the SDK. The implementation must also support a way to effectively cook the data. We begun using the UDP network stream since this approach was the easiest way to start using the system.

3.2.1 MVN Network Stream Unpacker in Max/MSP

A MXJ Java datagram unpacker was made for Max/MSP, but the implementation was shown to be too slow for real time applications. Though a dedicated Max external (in C++) would probably be faster, this architecture was not chosen for further development since Max/MSP does not, in our opinion, offer an effective data cooking environment.

3.2.2 Standalone Datagram Unpacker and Cooker

We wanted to continue using the Xsens Studio's UDP network stream, but with a more powerful data cooking environment. This was accomplished by implementing a standalone UDP datagram unpacking application. The programming language C++ was chosen since this is a fast and powerful computational environment. With this implementation we can either cook the data with self produced code or available libraries. Both raw and cooked data can then be sent as OSC messages for further cooking elsewhere or to the final sound engine.

3.2.3 Xsens MVN SDK Implementation

The Xsens MVN software development kit offers more data directly from the MVN engine compared to the UDP network stream. In addition to position, we get: positional and angular acceleration, positional and angular velocity and information about the sensor's magnetic disturbance. Every

[2]Most Ethernet network cards support 1500 bytes. Those supporting Jumbo frames can support up to 9000 bytes.

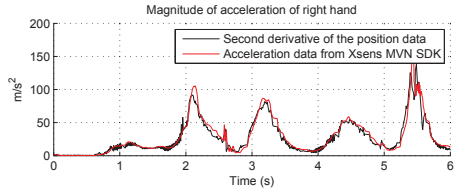


Figure 2: Difference between the second derivative of the position data versus the acceleration data obtained directly from MVN engine (SDK).

time frame is also marked with a time stamp that can be useful for analysis and synchronizing. Another benefit is that we have more control since we are directly communicating with the MVN engine and not listening for UDP packages. The drawback with the SDK is that we lose the benefit of using the user friendly MVN Studio and its GUI.

We implemented a terminal application with the SDK, that supports the basic Xsens features (calibration, playback, etc.). Since the application is getting data directly from the MVN engine we can save network overhead by cooking them in the same application before sending them as OSC messages. We also implemented a function that can send the motion data in the same data format as the Network UDP Datagram stream. This stream can then be opened by MVN Studio to get real-time visual feedback of the MoCap data.

3.2.4 Discussion

Since the solution presented in 3.2.2 offered a fast environment for data cooking, and let us use the user friendly MVN Studio, we have mainly used this approach in our work. We later discovered that the network stream offered by MVN Studio suffers from frame loss when driven in live mode, which affects both solutions presented in 3.2.1 and 3.2.2. Because of this we plan to focus on our SDK implementation in the future. An added advantage is that we no longer need to differentiate the segments positional data to be able to get properties like velocity and acceleration, since the SDK offers this directly from the MVN Engine. These data, especially the acceleration, seems to be of a higher quality since they are computed directly on the basis of the Xsens sensors and not differentiated from estimated position data as shown in Figure 2.[3]

3.3 Cooking Full Body MoCap Data

The Xsens MVN offers a wide range of different data to our system. If we use the network stream from the MVN Studio, each frame contains information about the position and orientation of 23 body segments. This yields in total 138 floating points numbers at a rate of 120Hz. Even more data will be available if one instead uses the MVN SDK as the source. Also different transformations and combinations of the data can be of interest, such as calculating distances or angles between body limbs.

Furthermore, we can differentiate all the above mentioned data to get properties like velocity, acceleration and jerk. Also, filters can be implemented to get smoother data or to emphasize certain properties. In addition, features like quantity of motion or "energy" can be computed. And with pattern recognition techniques we have the potential to recognize even higher level features [8].

We are currently investigating the possibilities that the Xsens MVN suit provides for musical interaction, but the mapping discussion is out of scope for this paper. Nevertheless, we believe it is important to be aware of the characteristics of the data we are basing our action-sound mappings on. We will therefore present technical performance details of the Xsens MVN system in the following section.

[3]The systems that tries to minimize positional drift probably contributes to a mismatch between differentiated positional data and the velocity and acceleration data from the MVN engine.

4. PERFORMANCE

4.1 Latency in a Sound Producing Setup

To be able to measure the typical expected latency in a setup like that of Figure 1 we performed a simple experiment with an audio recorder. One laptop was running our SDK implementation and sent OSC messages containing the acceleration of the hands. A patch in Max/MSP was made that would trigger a simple impulse response if the hands' acceleration had a high peak, which is a typical sign of two hands colliding to a sudden stop. The time difference between the acoustic hand clap and the triggered sound should then indicate the typical expected latency for the setup.

The Max/MSP patch was in experiment 1 running on the same laptop[4] as the SDK. In experiment 2 the patch was run on a separate Mac laptop[5] and received OSC messages through a direct Gbit Ethernet link. Experiment 3 was identical to 2 except that the Mac was replaced with a similar Windows based laptop. All experiments used the same firewire soundcard, *Edirol FA-101*. The results are given in Table 1 and are based on 30 measurements each which was manually examined in audio software. The standard deviation is included as an indication of the jitter performance. We can conclude that experiment 2 has the fastest sound output response while experiments 1 and 3 indicate that the Ethernet link did not contribute to a large amount of latency.

The Xsens MVN system offers a direct USB connection as an option for the Bluetooth wireless link. We used this option in experiment 4, which was in other ways identical to experiment 2. The results indicate that the direct USB connection is around 10-15 milliseconds faster and has a lower jitter performance than the Bluetooth link.

The upper boundary for "intimate control" has been suggested to be 10ms for latency and 1ms for its variations (jitter) [12]. If we compare the boundary with our results, we see that overall latencies are too large and that the jitter performance is even worse. However, in our experience, the system is still usable in many cases dependent on the designed action-sound mappings.

Table 1: Statistical results of the measured action to sound latency, in milliseconds.

Experiment	min	mean	max	std. dev.
1 Same Win laptop	54	66.7	107	12.8
2 OSC to Mac	41	52.2	83	8.4
3 OSC to Win	56	68	105	9.8
4 OSC to Mac - USB	28	37.2	56	6.9

4.2 Frame Loss in the Network Stream

We discovered that the Xsens MVN Studio's (version 2.6 and 3.0) network stream is not able to send all frames when running at 120Hz in real time mode on our computer.[3] At this rate it is skipping 10 to 40 percent of the frames. This does not need to be a significant problem if one use "time independent" analysis, that is analysis that does not look at the history of the data. But if we perform differential calculations on the Xsens data streams, there will be large jumps

[4]Dell Windows 7.0 Intel i5 based laptop with 4GB RAM
[5]MacBook Pro 10.6.6, 2.66 GHz Duo with 4GB RAM

Proceedings of the International Conference on New Interfaces for Musical Expression, 30 May - 1 June 2011, Oslo, Norway

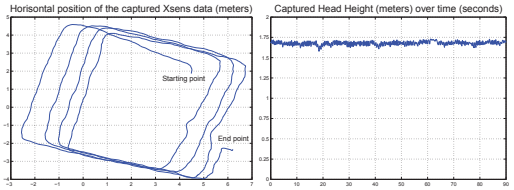


Figure 3: Plots of the captured horizontal (left) and vertical (right) position of the head.

in differentiated values during lost frames, hence noise. This was partly dealt with in the implementation described in 3.2.2. Whenever frames are detected as missing, the software will perform an interpolation. However, frame loss is still a major problem since we are not getting all the motion capture data and can lose important details in the data stream. For instance, if a trigger algorithm is listening for some sudden action, a couple of lost frames can make the event unrecognisable.

4.3 Positional Drift

The sensors in the Xsens MVN suit can only observe relative motion and calculate position through integration. This introduces drift. To be able to observe this drift we conducted a simple test by letting a subject walk along a rectangular path (around 6x7 meters) four times. Figure 3 shows a horizontal positional drift of about 2 meters during the 90 second long capture session. We can therefore conclude that Xsens MVN is not an ideal MoCap system if absolute horizontal position is needed.[6] The lack of drift in the vertical direction however, as can be seen in the right plot in Figure 3, is expected since the MVN engine maps the data to a human body model and assumes a fixed floor level.

4.4 Floor Level

If the motion capture area consists of different floor levels, like small elevated areas, the MVN engine will match the sensed raw data from the suit against the floor height where the suit was calibrated. This can be adjusted for in the post processing, but the real-time data will suffer from artifacts during floor level changes.

4.5 Magnetic Disturbance

The magnetic disturbance is critical during the calibration process but does not, to our experience, alter the motion tracking quality dramatically. During a concert we experienced significant magnetic disturbance, probably because of the large amount of electrical equipment on stage. But this did not influence the quality of MoCap data in such a way that it altered our performance.

4.6 Wireless Link Performance

Xsens specifies a maximum range up to 150 meters in an open field [13]. In our experience the wireless connection can easily cover an area with a radius of more than 50 meters in open air. Such a large area cannot be practically covered using IrMoCap systems.

We have performed concerts in three different venues.[7] During the two first concerts we experienced no problems with the wireless connection. During the third performance we wanted to test the wireless connection by increasing the distance between the Xsens suit and the receivers to about 20 meters. The wireless link also had an added challenge since the concert was held in a conference venue where we expected constant WIFI traffic. This setup resulted in problems with the connection and added latency. The distance should therefore probably be minimized when performing in venues with considerable wireless radio traffic.

[6]The product *MVN MotionGrid* will improve this drift.

[7]First concert: `www.youtube.com/watch?v=m1OffxIArrAi`

4.7 Final Performance Discussion

We believe that the Xsens MVN suit, in spite of its shortcomings in latency, jitter and positional drift, offers useful data quality for musical settings. However, the reported performance issues should be taken into account when designing action-sound couplings. We have not been able to determine whether the Xsens MVN system preserves the motion qualities we are most interested in compared to other MoCap systems, nor how their performance compares in real life settings. To be able to answer more of these questions we are planning systematic experiments comparing Xsens MVN with other MoCap technologies.

5. FUTURE WORK

In Section 3.3 we briefly mentioned the vast amount of data that is available for action-sound mappings. Not only are there many possibilities to investigate, it also involves many mathematical and computational details. However, the challenges associated with the cooking of full body MoCap data are not specific to the Xsens MVN system. Other motion capture systems like IrMoCap systems offer similar data. It should therefore be profitable to make one cooking system that can be used for several MoCap technologies.

The main idea is to gather effective and fast code for real time analysis of motion capture data; not only algorithms but also knowledge and experience about how to use them. Our implementation is currently specialized for the the Xsens MVN suit. Future research includes incorporating this implementation with other motion capture technologies and develop a real time motion capture toolbox.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/motion_capture.

[2] T. Cloete and C. Scheffer. Benchmarking of a full-body inertial motion capture system for clinical gait analysis. In *EMBS*, pages 4579 –4582, 2008.

[3] N. Collins, C. Kiefer, Z. Patoli, and M. White. Musical exoskeletons: Experiments with a motion capture suit. In *NIME*, 2010.

[4] R. Dannenberg. *Real-time scheduling and computer accompaniment*. MIT Press, 1989.

[5] V. Lympourides, D. K. Arvind, and M. Parker. Fully wireless, full body 3-d motion capture for improvisational performances. In *CHI*, 2009.

[6] V. Lympouridi, M. Parker, A. Young, and D. Arvind. Sonification of gestures using specknets. In *SMC*, 2007.

[7] P.-J. Maes, M. Leman, M. Lesaffre, M. Demey, and D. Moelants. From expressive gesture to sound. *Journal on Multimodal User Interfaces*, 3:67–78, 2010.

[8] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis. A gesture-driven multimodal interactive dance system. In *ICME*, 2004.

[9] D. Rosenberg, H. Luinge, and P. Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Technologies*, 2009.

[10] A. Schmeder, A. Freed, and D. Wessel. Best practices for open sound control. In *LAC*, 2010.

[11] S. Skogstad, A. R. Jensenius, and K. Nymoen. Using ir optical marker based motion capture for exploring musical interaction. In *NIME*, 2010.

[12] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *NIME*, 2001.

[13] Xsens Technologies B.V. *Xsens MVN User Manual*.

Paper III

Comparing Inertial and Optical MoCap Technologies for Synthesis Control.
S.A. Skogstad, K. Nymoen, and M.E. Høvin.
In *Proceedings of SMC 2011 8th Sound and Music Computing Conference "Creativity rethinks science"*, pages 421–426, Padova University Press 2011.

COMPARING INERTIAL AND OPTICAL MOCAP TECHNOLOGIES FOR SYNTHESIS CONTROL

Ståle A. Skogstad and Kristian Nymoen
fourMs - Music, Mind, Motion, Machines
Department of Informatics
University of Oslo
{savskogs,krisny}@ifi.uio.no

Mats Høvin
Robotics and Intelligent Systems group
Department of Informatics
University of Oslo
matsh@ifi.uio.no

ABSTRACT

This paper compares the use of two different technologies for controlling sound synthesis in real time: the infrared marker-based motion capture system *OptiTrack* and *Xsens MVN*, an inertial sensor-based motion capture suit. We present various quantitative comparisons between the data from the two systems and results from an experiment where a musician performed simple musical tasks with the two systems. Both systems are found to have their strengths and weaknesses, which we will present and discuss.

1. INTRODUCTION

Motion capture (MoCap) has become increasingly popular among music researchers, composers and performers [1]. There is a wide range of different MoCap technologies and manufacturers, and yet few comparative studies between the technologies have been published. Where one motion capture technology may outperform another in a sterilized laboratory setup, this may not be the case if the technologies are used in a different environment. Optical motion capture systems can suffer from optical occlusion, electromagnetic systems can suffer from magnetic disturbance, and so forth. Similarly, even though one motion capture system may be better than another at making accurate MoCap recordings and preparing the motion capture for offline analysis, the system may not be as good if the task is to do accurate motion capture in real time, to be used for example in controlling a sound synthesizer.

In this paper we compare the *real-time* performance of two motion capture systems (Figure 1) based on different technologies: Xsens MVN which is based on inertial sensors, and OptiTrack which is an infrared marker-based motion capture system (IrMoCap). Some of our remarks are also relevant to other motion capture systems than the ones discussed here, though the results and discussions are directed only toward OptiTrack and Xsens.

We will return to a description of these technologies in section 3. In the next section we will give a brief overview of related work. Section 4 will present results from comparisons between the two motion capture systems, which are then discussed in section 5.

Figure 1. The NaturalPoint OptiTrack system (left) and the Xsens MVN system (right).

2. RELATED WORK AND BACKGROUND

Motion capture technologies have been used in musical contexts for a long time, and during the 00's we saw several examples of using various motion capture technologies for real-time control of sound. This includes electromagnetic motion capture [2], video-based motion capture [3], optical marker-based motion capture [4] and inertial motion capture [5], to mention a few.

Several researchers have reported on differences between motion capture technologies. Most of these reports, however, have been related to offline analysis for medical or animation purposes. Cloete et al. [6] have compared the kinematic reliability of the Xsens MVN suit with an IrMoCap system during routine gait studies. They conclude that the Xsens MVN system is comparable to IrMoCap systems but with shortcomings in some angle measurements. They also point out several practical advantages with the Xsens suit, like its wireless capabilities and quick set-up time. Another experiment by Thies et al. [7] found comparable acceleration values from two Xsens sensors and an IrMoCap system, and showed that calculating acceleration from the IrMoCap position data introduced noise. One of the conclusions from this experiment was that filtering methods need to be investigated further.

Miranda and Wanderley have pointed out some strengths and weaknesses with electromagnetic and optical motion capture systems [1]: Electromagnetic systems are able to track objects, even if it is not within the direct line of sight of external cameras. On the other hand, these systems need cables which may be obtrusive. Optical systems are superior to many other systems in terms of sampling rate, since they may track markers at sampling rates of more than 1000 Hz, and systems using passive markers have no need for obtrusive cables. Still, these systems need a direct line of sight between markers and cameras, and a passive

marker system may not be able to uniquely identify each marker.

Possibilities, strengths and weaknesses for real-time motion capture in musical contexts are discussed individually for IrMoCap and full-body inertial sensor systems in [8] and [9]. In this paper we will compare the real-time abilities of the two technologies.

2.1 Initial remarks on requirements when using MoCap for real-time control of music

A musical instrument is normally controlled with excitation and modification actions [10]. We can further distinguish between two types of excitations: discrete (i.e. trigger), or continuous (like bowing a string instrument). Dobrian [11] identifies two types of control data: triggers and streams of discrete data representing a sampling of a continuous phenomenon. Following these remarks, we are looking for a system able to robustly trigger sound events with good temporal accuracy, and to continuously control a system with good spatial accuracy and little noise. Consequently, we have chosen to emphasize three properties: spatial accuracy, temporal accuracy and system robustness. We will come back to measurements and discussion of these properties in sections 4 and 5.

3. TECHNOLOGIES

3.1 NaturalPoint OptiTrack

NaturalPoint OptiTrack is an optical infrared marker-based motion capture system (IrMoCap). This technology uses several cameras, equipped with infrared light-emitting diodes. The infrared light from the cameras is reflected by reflective markers and captured by each camera as 2D point-display images. By combining several of these 2D images the system calculates the 3D position of all the markers within the capture space. A calibration process is needed beforehand to determine the position of the cameras in relationship to each other, and in relationship to a global coordinate system defined by the user.

By using a combination of several markers in a specific pattern, the software can identify rigid bodies or skeletons. A *rigid body* refers to an object that will not deform. By putting at least 3 markers on the rigid body in a unique and non-symmetric pattern, the motion capture system is able to recognize the object and determine its position and orientation. A *skeleton* is a combination of rigid bodies and/or markers, and rules for how they relate to each other. In a human skeleton model, such a rule may be that the bottom of the right thigh is connected to the top of the right calf, and that they can only rotate around a single axis. In the NaturalPoint motion capture software (Arena), there exist 2 predefined skeleton models for the human body. It is not possible to set up user-defined skeletons.

3.2 The Xsens MVN

The Xsens MVN technology can be divided into two parts: (1) the sensor and communication hardware that are responsible for collecting and transmitting the raw sensor data, and (2) the Xsens MVN software engine, which interprets and reconstructs the data to full body motion while trying to minimize positional drift.

The Xsens MVN suit [12] consists of 17 inertial MTx sensors, which are attached to key areas of the human body. Each sensor consists of 3D gyroscopes, accelerometers and magnetometers. The raw signals from the sensors are connected to a pair of Bluetooth 2.0-based wireless transmitters, which again transmit the raw motion capture data to a pair of wireless receivers.

The data from the Xsens MVN suit is fed to the MVN software engine that uses sensor fusion algorithms to produce absolute orientation values, which are used to transform the 3D linear accelerations to global coordinates. These in turn are translated to a human body model which implements joint constraints to minimize integration drift. The Xsens MVN system outputs information about body motion by expressing body postures sampled at a rate up to 120Hz. The postures are modeled by 23 body segments interconnected with 22 joints.

4. MEASUREMENTS

We carried out two recording sessions to compare the OptiTrack and Xsens systems. In the first session, a series of simple measurements were performed recording the data with both Xsens and OptiTrack simultaneously. These recordings were made to get an indication of the differences between the data from the systems. In the second session (Section 4.5), a musician was given some simple musical tasks, using the two MoCap systems separately to control a sound synthesizer.

4.1 Data comparison

Our focus is on comparing real-time data. Therefore, rather than using the built-in offline recording functionality in the two systems, data was streamed in real-time to a separate computer where it was time-stamped and recorded. This allows us to compare the quality of the data as it would appear to a synthesizer on a separate computer. Two terminal applications for translating the native motion capture data to Open Sound Control and sending it to the remote computer via UDP were used.

We have chosen to base our plots on the unfiltered data received from the motion capture systems. This might differ from how a MoCap system would be used in a real world application, where filtering would also be applied. Using unfiltered data rather than filtered data gives an indication of how much pre-processing is necessary before the data can be used for a musical application.

The Xsens suit was put on in full-body configuration. For OptiTrack, a 34-marker skeleton was used. This skeleton model is one of the predefined ones in the Arena software. Markers were placed outside the Xsens suit, which made it necessary to adjust the position of some of the markers slightly, but this did not alter the stability of the OptiTrack system.

Both systems were carefully calibrated, but it was difficult to align their global coordinate systems perfectly. This

is because OptiTrack uses a so-called L-frame on the floor to determine the global coordinate system, whereas Xsens uses the position of the person wearing the suit during the calibration to determine the origin of the global coordinate system. For this reason, we get a bias in the data from one system compared to the other. To compensate for this, the data has been adjusted so that the mean value of the data from the two systems more or less coincide. This allows us to observe general tendencies in the data.

4.2 Positional accuracy and drift

When comparing the Xsens and the OptiTrack systems there is one immediately evident difference. OptiTrack measures absolute position, while the sensors in the Xsens MVN suit can only observe relative motion. With Xsens, we are bound to experience some positional drift even though the system has several methods to keep it to a minimum [9].

4.2.1 Positional accuracy - still study

Figure 2 shows the position of the left foot of a person sitting in a chair without moving for 80 seconds. The upper plot shows the horizontal (XY) position and the lower plot shows vertical position (Z) over time. In the plot it is evident that Xsens suffers from positional drift, even though the person is sitting with the feet stationary on the floor. Xsens reports a continuous change of data, with a total drift of more than 0.2 m during the 80 seconds capture session. Equivalent plots of other limbs show similar drift, hence there is little relative drift between body limbs.

This measurement shows that OptiTrack is better at providing accurate and precise position data in this type of clinical setup. However, for the vertical axis, we do not observe any major drift, but the Xsens data is still noisier than the OptiTrack data.

4.2.2 Positional accuracy - walking path

The left plot in Figure 3 displays the horizontal (XY) position of the head of a person walking along a rectangular path in a large motion capture area recorded with Xsens. The plot shows a horizontal positional drift of about 2 meters during the 90 seconds capture session. Xsens shows

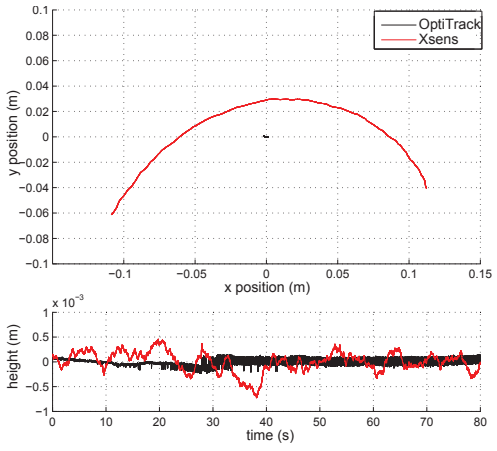


Figure 2. Horizontal and vertical plots of a stationary foot.

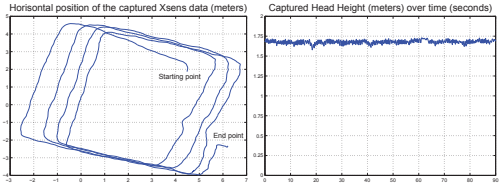


Figure 3. Recording of the horizontal (left) and vertical (right) position of the head.

no drift in the vertical direction (Z), as can be seen in the right plot. This is expected since the MVN engine maps the data to a human body model and assumes a fixed floor level. Because of the major horizontal drift we can conclude that Xsens MVN is not an ideal MoCap system if absolute horizontal position is needed.

4.2.3 Camera occlusion noise

The spatial resolution of an IrMoCap system mainly relies on the quality of the cameras and the calibration. The cameras have a certain resolution and field of view, which means that the spatial resolution of a marker is higher close to the camera than far away from the camera. The calibration quality determines how well the motion capture system copes with the transitions that happen when a marker becomes visible to a different combination of cameras. With a "perfect" calibration, there might not be a visible effect, but in a real situation we experience a clearly visible change in the data whenever one or more cameras fail to see the marker, as shown in Figure 4. When a marker is occluded from a camera, the 3D calculation will be based on a different set of 2D images.

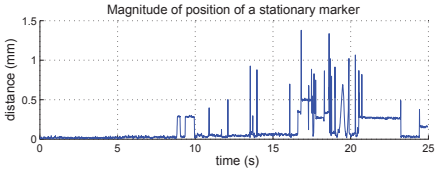


Figure 4. OptiTrack: Magnitude of the distance from the mean position of a stationary marker. The disturbances in the last part of the measurement is caused when a person moves around the marker, and thus blocks the marker in one or more cameras at a time. FrameRate 100 Hz

4.2.4 Xsens floor level change

If the motion capture area consists of different floor levels, like small elevated areas, the Xsens MVN engine will match the sensed raw data from the suit against the floor height where the suit was calibrated. This can be adjusted in post-processing, but real-time data will suffer from artifacts during floor level changes, as shown in Figure 5.

4.3 Acceleration and velocity data

In our experience, velocity and acceleration are highly usable motion features for controlling sound. High peaks in absolute acceleration can be used for triggering events, while velocity can be used for continuous excitation.

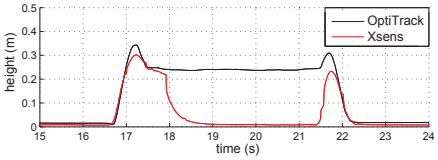


Figure 5. Recording of the vertical position of the left foot of a person, stepping onto an elevated area (around 0.25 m high). When the user plants his left foot on the object, the Xsens MVN engine will eventually map the stationary foot to floor level (18 to 19 s).

A difference between the two MoCap systems is that the Xsens system can offer velocity and acceleration data directly from the MVN engine [9]. When using the OptiTrack system we need to differentiate position data to estimate velocity and acceleration. If the positional data is noisy, the noise will be increased by differentiation (act as an high-pass filter), as we can see from Figure 6. The noise resulting from optical occlusion (see Section 4.2.3) is probably the cause for some of OptiTrack's positional noise.

Even though the Xsens position data is less accurate, it does offer smoother velocity and, in particular, acceleration data directly. We can use filters to smooth the data from the OptiTrack system; however, this will introduce a system delay, and hence increased latency.

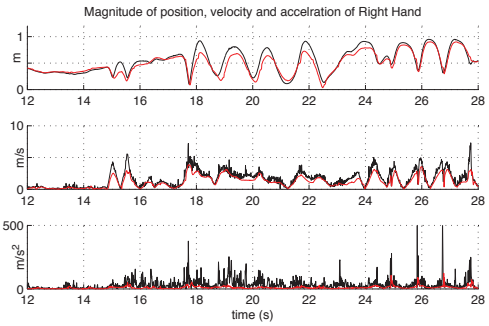


Figure 6. Velocity and acceleration data quality comparison (OptiTrack in black and Xsens in red).

4.4 Action-to-sound: latency and jitter

Low and stable latency is an important concern for *real-time* musical control [13], particularly if we want to use the system for triggering temporally accurate musical events. By *action-to-sound latency* we mean the time between the sound-producing action and the sonic reaction from the synthesizer.

To be able to measure the typical expected latency in a setup like that in Figure 7 we performed a simple experiment with an audio recorder. One computer was running one of the MoCap systems and sent OSC messages containing the MoCap information about the user's hands. A patch in Max/MSP was made that registered hand claps

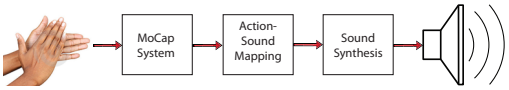


Figure 7. The acoustic hand clap and the triggered sound were recorded to measure latency of the systems.

based on MoCap data and triggered a *click* sound for each clap. The time difference between the acoustic hand clap and the triggered sound should indicate the typical expected latency for the setup.

Both MoCap systems were run on the same PC [1]. The sound-producing Max/MSP patch was run on a separate Mac laptop [2] and received OSC messages from the MoCap systems through a direct Gbit Ethernet link. All experiments used the same firewire connected sound card, *Edirol FA-101*, as output source. The hand claps and the click output from the Max patch was recorded with a microphone. Statistical results from the time delays between hand claps and corresponding click sound in the recorded audio files are given in Table 1. The values are based on 30 claps each. In this experiment, OptiTrack had a faster sound output response and a lower standard deviation than Xsens. The standard deviation is included as an indication of the jitter performance of the MoCap systems, since lower standard deviation indicates higher temporal precision.

Higher Xsens latency and jitter values are probably partly due to its use of Bluetooth wireless links. The Xsens MVN system also offers a direct USB connection option. We performed the same latency test with this option; and the results indicate that the connection is around 10-15 milliseconds faster, and has a lower jitter performance, than the Bluetooth link.

The upper bounds for "intimate control" have been suggested to be 10ms for latency and 1ms for its variations (jitter) [13]. If we compare the bounds with our results, we see that both systems have relatively large latencies. However, in our experience, a latency of 50ms is still usable in many cases. The high jitter properties of the Xsens system are probably the most problematic, especially when one wants high temporal accuracy.

	min	mean	max	std. dev.
OptiTrack	34	42.5	56	5.0
Xsens Bluetooth	41	52.2	83	8.4
Xsens USB	28	37.2	56	6.9

Table 1. Statistical results of the measured action-to-sound latency, in milliseconds.

4.5 Synthesizer control

In a second experiment, a musician was asked to perform simple music-related tasks with the two motion capture

[1] Intel 2.93 GHz i7 with 8GB RAM running Win 7
[2] MacBook Pro 10.6.6, 2.66 GHz Duo with 8GB RAM

systems. Three different control mappings to a sound synthesizer were prepared:

- Controlling pitch with the distance between the hands
- Triggering an impulsive sound based on high acceleration values
- Exciting a sustained sound based on the velocity of the hand

For the pitch mapping, the task was to match the pitch of one synthesizer to the pitch of another synthesizer moving in the simple melodic pattern displayed in Figure 8, which was repeated several times. This task was used to evaluate the use of position data from the two systems as the control data.

For the triggering mapping, the task was to follow a pulse by clapping the hands together. This task was given to evaluate acceleration data from the two systems as the control data, and to see if the action-to-sound latency and jitter would make it difficult to trigger events on time.

The excitation mapping was used to follow the loudness of a synthesizer, which alternated between ”on” and ”off” with a period of 1 second. This task was used to evaluate velocity data as control data.

The *reference sound* (the sound that the musician was supposed to follow) and the *controlled sound* (the sound that was controlled by the musician) were played through two different loudspeakers. The two sounds were also made with different timbral qualities so that it would be easy to distinguish them from each other. The musician was given some time to practice before each session. To get the best possible accuracy, both systems were used at their highest sampling rates for this experiment: Xsens at 120 Hz, and OptiTrack at 100 Hz.

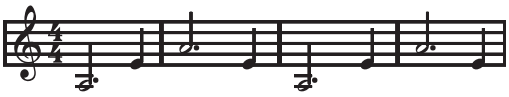


Figure 8. The simple melody in the pitch-following task. This was repeated for several iterations.

4.5.1 Pitch-following results

We found no significant difference between the performances with the two systems in the pitch-following task. Figure 9 displays an excerpt of the experiment, which shows how the participant performed with both Xsens and OptiTrack. The participant found this task to be difficult, but not more difficult for one system than the other. Also, the data shows no significant difference in the performances with the two systems. This indicates that the quality of relative position values (between markers/limbs) is equally good in the two systems for this kind of task.

4.5.2 Triggering results

Table 2 shows the results of the latency between the reference sound and the controlled sound for the triggering test. They are based on 40 hand claps for each of the two

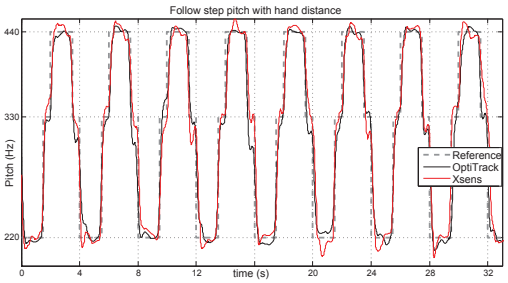


Figure 9. There was no significant difference between the two systems for the pitch-following task.

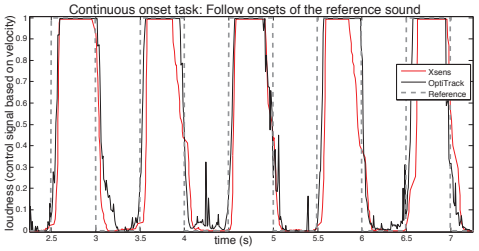


Figure 10. The major difference between the two systems in the continuous onset task was the noisy data from the OptiTrack system, which made it difficult to be quiet between the onsets. Apart from this, there was no big difference between the two systems.

MoCap systems. As we can see, the *mean* latency value is almost equal for Xsens and OptiTrack. Xsens has a higher standard deviation, which may indicate that the Xsens jitter shown in Table 1 makes it difficult for the user to make a steady trigger pulse.

	min	mean	max	std. dev.
OptiTrack	18.5	45.2	77.1	13.8
Xsens	2.6	44,7	96.3	28.3

Table 2. Statistical results, in milliseconds, of the measured time differences between reference signal and control signal.

4.5.3 Continuous onset results

For the continuous onset task, where the loudness of the sound was controlled by the absolute velocity of the right hand, we also observed a time delay between the onset of the reference tone and the onset of the sound played by our performer. This delay was present for both systems. In this task, the OptiTrack system suffered from noise, which was introduced when calculating the absolute velocity of the unfiltered OptiTrack data, as described in Section 4.3 (see Figure 10). The musician said that this made it more difficult to be quiet between the reference tones, and that this task was easier to perform with the Xsens system.

5. DISCUSSION

We have seen several positive and negative aspects with the quantitative measurements of the two technologies. In this section we will summarize our experiences of working with the two systems in a music-related context.

The main assets of the Xsens suit is its portability and wireless capabilities. The total weight of the suit is approximately 1.9 kg and the whole system comes in a suitcase with the total weight of 11 kg. Comparably, one could argue that a 8-camera OptiTrack setup could be portable, but this system requires tripods, which makes it more troublesome to transport and set up. OptiTrack is also wireless, in the sense that the user only wears reflective markers with no cables, but the capture area is restricted to the volume that is covered by the cameras, whereas Xsens can easily cover an area with a radius of more than 50 meters. When designing a system for real-time musical interaction based on OptiTrack, possible marker dropouts due to optical occlusion or a marker being moved out of the capture area must be taken into account. For Xsens, we have not experienced complete dropouts like this, but the Bluetooth link is vulnerable in areas with heavy wireless radio traffic, which may lead to data loss. Nevertheless, we consider Xsens to be the more robust system for on-stage performances.

OptiTrack has the benefit of costing less than most other motion capture technologies with equivalent resolution in time and space. The full Xsens suit is not comfortable to wear for a longer time period, whereas OptiTrack markers impose no or little discomfort. On the other hand, OptiTrack markers can fall off when tape is used to attach them. Also, OptiTrack's own solution for hand markers, where a plastic structure is attached to the wrist with Velcro, tends to wobble a lot, causing very noisy data for high acceleration movement, something we experienced when we set up the hand clapping tests. Xsens has a similar problem with the foot attachments of its sensors, which seems to cause positional artifacts.

Sections 4.2 to 4.5 show a number of differences between Xsens and OptiTrack. In summary, OptiTrack offers a higher positional precision than Xsens without significant drift, and seemingly also lower latency and jitter. Xsens delivers smoother data, particularly for acceleration and velocity. Our musician subject performed equally well in most of the musical tasks. However, the noisy OptiTrack data introduced some difficulties in the continuous onset task, and also made it challenging to develop a robust algorithm for the triggering task. Furthermore, Xsens jitter made the triggering task more difficult for the musician.

6. CONCLUSIONS

Both OptiTrack and Xsens offer useful MoCap data for musical interaction. They have some shared and some individual weaknesses, and in the end it is not the clinical data that matters, but the intended usage. If high positional precision is required, OptiTrack is preferable over Xsens, but if acceleration values are more important, Xsens provide less noisy data without occlusion problems. Overall, we find Xsens to be the most robust and stage-friendly MoCap system for real-time synthesis control.

7. REFERENCES

[1] E. R. Miranda and M. Wanderley, *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, Inc., 2006.

[2] J. Michel Couturier and D. Arfib, "Pointing fingers: Using multiple direct interactions with visual objects to perform music," in *Proc . NIME*, 2003, pp. 184–188.

[3] G. Castellano, R. Bresin, A. Camurri, and G. Volpe, "Expressive control of music and visual media by full-body movement," in *Proc. NIME*. New York, USA: ACM, 2007, pp. 390–391.

[4] F. Bevilacqua, J. Ridenour, and D. J. Cuccia, "3d motion capture data: motion analysis and mapping to music," in *Proc. Workshop/Symposium SIMS*, California, Santa Barbara, 2002.

[5] P.-J. Maes, M. Leman, M. Lesaffre, M. Demey, and D. Moelants, "From expressive gesture to sound," *Journal on Multimodal User Interfaces*, vol. 3, pp. 67–78, 2010.

[6] T. Cloete and C. Scheffer, "Benchmarking of a full-body inertial motion capture system for clinical gait analysis," in *EMBS*, 2008, pp. 4579 –4582.

[7] S. Thies, P. Tresadern, L. Kenney, D. Howard, J. Goulermas, C. Smith, and J. Rigby, "Comparison of linear accelerations from three measurement systems during reach & grasp," *Medical Engineering & Physics*, vol. 29, no. 9, pp. 967 – 972, 2007.

[8] S. A. Skogstad, A. R. Jensenius, and K. Nymoen, "Using IR optical marker based motion capture for exploring musical interaction," in *Proc. NIME*, Sydney, Australia, 2010, pp. 407–410.

[9] S. A. Skogstad, K. Nymoen, Y. de Quay, and A. R. Jensenius, "Osc implementation and evaluation of the xsens mvn suit," in *Proc of NIME*, Oslo, Norway, 2011.

[10] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman, "Musical gestures: concepts and methods in research," in *Musical Gestures: Sound, Movement, and Meaning*, R. I. Godøy and M. Leman, Eds. New York: Routledge, 2010, pp. 12–35.

[11] C. Dobrian, "Aesthetic considerations in the use of 'virtual' music instruments," in *Proc. Workshop on Current Research Directions in Computer Music*, 2001.

[12] D. Rosenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," *Xsens Technologies*, 2009.

[13] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," in *Proc. NIME*, Seattle, USA, 2001.

Paper IV

Developing the Dance Jockey System for Musical Interaction with the Xsens MVN suit.
S.A. Skogstad, K. Nymoen, Y.d. Quay and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 226–229, University of Michigan 2012.

Developing the Dance Jockey System for Musical Interaction with the Xsens MVN Suit

Ståle A. Skogstad and Kristian Nymoen
University of Oslo, Department of Informatics
{savskogs,krisny}@ifi.uio.no

Yago de Quay
University of Porto, Faculty of Engineering
yagodequay@gmail.com

Alexander Refsum Jensenius
University of Oslo, Department of Musicology
a.r.jensenius@imv.uio.no

ABSTRACT

In this paper we present the *Dance Jockey System*, a system developed for using a full body inertial motion capture suit (Xsens MVN) in music/dance performances. We present different strategies for extracting relevant postures and actions from the continuous data, and how these postures and actions can be used to control sonic and musical features. The system has been used in several public performances, and we believe it has great potential for further exploration. However, to overcome the current practical and technical challenges when working with the system, it is important to further refine tools and software in order to facilitate making of new performance pieces.

1. INTRODUCTION

The Dance Jockey system is based on the Xsens MVN suit, a commercially available *full body* motion capture system. The suit consists of 17 inertial sensors that are attached to a pre-defined set of points on the human body. Each sensor consists of an accelerometer, a gyroscope, and a magnetometer. The raw data streams from these sensors are combined in the Xsens MVN system to produce an estimation of how the body moves [9].

In previous research we have shown that the Xsens MVN system is well suited for exploring full body musical interaction [9, 10]. The system offers robust motion tracking of the body, which is important in live performance settings. In [9] we presented the Open Sound Control implementation and the technical experience of using the Xsens MVN system. In this paper we will outline in more detail about how we used the Xsens MVN suit to control sonic and musical features in the *Dance Jockey* project (Figure 1).

The motivation for the Dance Jockey project came from our wish of using the full body for musical interaction. As is often commented on, performing with computers allows for many new and exciting sonic possibilities, but many times with a weak or missing connection between the actions of the performer and the output sound [1]. To overcome this problem of missing or unnatural action-sound couplings [6], we are trying to develop pieces in which properties of the output sound match properties of the performed actions. With Xsens MVN motion capture (MoCap) system we are able to measure, with some limitation, the physical properties of our bodies' actions. It should therefore be possible to use this data to create physical relationships between actions and sounds. The challenge, however, is to extract relevant features from the continuous motion capture data stream and turn these features into meaningful sound.

Figure 1: A Dance Jockey performance at Mostra UP in Porto, Portugal. Note the orange sensors on different body parts and the two wireless transmitters on the back of the performer.

The name Dance Jockey is a word play on the well-known term Disc Jockey, or DJ. With this name we wanted to reflect that instead of using discs to perform music, we were using dance or full body motion as the basis for the performance. The name is also a reference to how we may think of the performer more as a DJ/turntablist than a musician: the performer does not play an instrument with direct control of all sonic/musical features, he is more triggering and influencing various types of sonic material through his body.

The developed Dance Jockey System has been used in several public performances over the last years, many of which are documented on our project web page.[1] This paper will mainly focus on the system itself, and we will therefore not present and discuss the performances.

We will start by presenting the main structure of the Dance Jockey System, followed by an overview of different feature extraction methods that have been developed, and how they have been used to control sonic and musical features.

2. THE DANCE JOCKEY SYSTEM

The system on which we have based our Dance Jockey project can be divided into four main parts, as illustrated in Figure 2. Let us briefly look at the concept of sound excitation before presenting the features used to extract control signals.

[1] `http://www.fourms.uio.no/projects/dancejockey/`

NIME'12, May 21 – 23, 2012, University of Michigan, Ann Arbor.

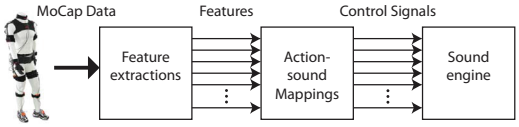


Figure 2: The dataflow of our Dance Jockey system

2.1 Sound Excitation

Most acoustic instruments are controlled with sound-producing actions that can be further broken into *excitation* and *modification* actions [7]. We can further distinguish between two types of excitations and modifications: discrete (e.g. triggering a sound object), or continuous (e.g. bowing a string instrument). This terminology can be seen as similar to what Dobrian identifies as control signals: *triggers* and *continuous streams of discrete data* [3]. These control signals should also be sufficient to control other musical features like tempo, skipping to the next section of the performance, changing synthesizer settings etc. Accordingly, we want to use the Xsens MVN data both for *continuous control* and to extract *trigger signals*.

2.2 Features Used for Extracting Control Signals

The Xsens MVN system outputs data about body motion by expressing body postures sampled at a rate of up to 120Hz. The postures are modeled by 23 body segments interconnected with 22 joints. Each posture sample consist of the *position* and the *orientation* of these segments. In addition, we get each segments' positional and orientational *velocity*, and positional and orientational *acceleration*. (The latter data are of relatively good quality as documented in [9].) All data is given in some *global* coordinate system, e.g. the stage.

There were three main properties we looked for when searching for suitable features from the above data; the features should be (1) robust and usable as consistent control data, (2) usable as visual cues for the audience, and (3) user-friendly for the artist. The features are difficult to evaluate without considering how they are mapped to musical parameters. It is therefore important to include the typical use of the features in the following subsections. We have not tried to make a complete list of all available features; instead, we will present those that we found useful. The features are summarized in Table 1 and several examples are illustrated in Figure 3.

2.2.1 Position data

We could, in theory, use the segments' global positions for both continuous control and extracting triggers by placing virtual positional thresholds on the stage (Figure 3e). But, we did not use the global position directly since the Xsens MVN *horizontal* position data exhibits drift, as documented in [9]. The vertical position, however, is much more consistent and could therefore be used directly as a feature. The latter can also be seen as a global feature since, for example, 1 meter above floor level will stay the same in all parts of the stage (Figure 3a).

The possibility of using global positions for sound spatialization is interesting. However, using global horizontal position for other types of sound excitation is somewhat problematic. We wanted actions in one area of the stage to result in the same output in other areas of the stage. In order to achieve this, we transformed global positions to the local coordinate system of the performer (pelvis). A specified action would then result in the same output in all areas of the stage, regardless of the orientation or position of the performer. This technique is also immune to the Xsens positional drift problem to a large extent. We used this approach when placing virtual "wind chimes" around the performer, who was able to trigger chimes by touching these virtual positions without worrying about standing in the correct position on the stage (Figure 3f).

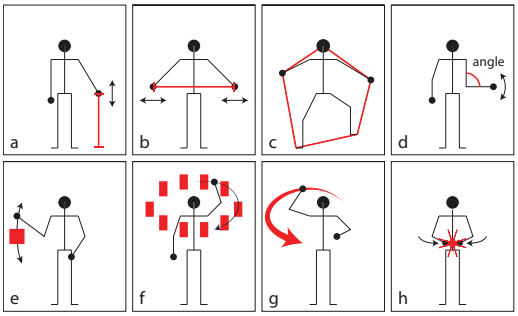


Figure 3: Illustration of some of the different features we have used: (a) vertical height of a hand, (b) distance between hands, (c) spanned distance between main body limbs, (d) elbow angle, (e) virtual trigger area, activated when the hand passes through this area, (f) virtual trigger areas that are always relative to the performer (g) absolute speed of hand, and (h) thresholding acceleration to recognize a hand clap.

2.2.2 Velocity - Continuous Excitation

We found the positional velocity of body limbs, especially the absolute velocity, i.e. the magnitude of velocity in all 3 dimensions, to be especially useful for continuous excitations (follows what Hunt et. al. discovered in [5]). This can also be mapped in an intuitive way with the performer's physical effort: the faster/larger the movement, the louder the sound. A benefit of using absolute velocity is its global nature: it is based on total velocity of the moving limb and is independent of the direction or location of the motion. We used this feature mostly for continuous control, for instance controlling amplitude or filters (Figure 3g).

2.2.3 Acceleration - Triggers

We found *thresholding* acceleration values to be especially suitable for extracting trigger signals, which is also mentioned by Bevilacqua et. al. in [2]. For example, the performer was able to trigger sound samples via abrupt rotations of his hand by thresholding the *rotational* acceleration data. We also used the performer's hip rotations to trigger samples. In this way we were able to synchronize sounds with apparent dance actions.

One of the challenges of using acceleration for extracting triggers is that sudden motion in one part of the body often spread to other parts of the body. As a consequence, it was difficult to isolate different triggers from each other, e.g. separating a kick from a sudden hip movement when only thresholding the segments' acceleration values. We overcame this by specifying extra *conditions* for the different trigger algorithms that needed to be separated. For instance, to be able to safely trigger a hand clap we added the condition that the hands needed to be no more than 20 cm from each other (Figure 3h). In this way we were able

to avoid other abrupt hand movement resulting in "hand clap" triggers. In similar ways we can make appropriate conditions for other trigger algorithms, such that they only trigger by the specified body action. This is one of the benefits of using a full body MoCap system (compared to using single accelerometers).

2.2.4 Quantity of motion (QoM)

By summing up the speeds of different body limbs we can compute the performer's total *quantity of motion*. To save computational power, we can add up the speed of only a subset of the main limbs, like head, feet and hands. This gives similar results. We connected this feature to loudness and other effort-related associations in the sound output, and we believe it is an interesting higher-level motion feature. However, the performer found this feature to be difficult to consciously control (low repeatability), and we therefore found it as having only limited use for extracting control signals.

2.2.5 Relative position between body segments

The Xsens MVN system outputs data which is mapped to a human body model. We find this model to be quite consistent and stable and therefore an interesting source for extracting control signals. It does not suffer from optical occlusion like infra-red optical marker based motion capture systems or have other major noise sources [9]. We do however experience some limited drift between limbs, but if this drift is taken into account the relations between different body parts can in our experience be quite robust and useful. (This property also applies for subsections 2.2.6 and 2.2.7.)

As a simple example, we used the distance between the performer's hands to reflect a physical space that the performer could manipulate, which again was used to make a physical relationship with the output sound (Figure 3b). Another feature that we used was the spanned distance of the 5 main body extremities: head, hands and feet. We used this distance for continuous excitation and modification, and found it useful to excite sound in a visually dramatic way (Figure 3c).

2.2.6 Orientations - Joint Angles

We did not use the segments orientation data directly. Instead, we used them to calculate the angles between different segments to extract joint angles, e.g. elbows and knees (Figure 3d). We believe that joint angles are more useful features than using the global orientation of single body limbs, since they tell more about the body pose. These angles are also relative to the performer's body. We used them to continuously excite or modify sound(s), and thresholded them to extract trigger signals.

2.2.7 Pose classifier

We developed a simple recognition algorithm based on an idea that different body poses could control some aspects of the sounds, besides also being valuable visual cues for communicating with the audience. We picked out five key pose features: the two elbow angles, hand distance, and both hand heights. Together these features spanned a pose space in five dimensions. We then stored the corresponding features of a set of 9 poses (the one we wanted to use as "cues" or "control poses"). These poses then had a corresponding point in the pose space. Finally, we implemented a *Nearest Neighbor Classifier* [4] to classify poses to the one of the stored poses that was closest, see Figure 4 for an illustration.

An advantage of this classifier was the high recognition rate, which in practice was 100%. This made it useful for exciting important musical features like notes and chords. However, the performer had problems with timing the pose changes correctly. To overcome this we implemented a system where a metronome was responsible for triggering the pose changes. In this way the performer only needed to be in the right pose at the right time. We also implemented functionality that looked after certain sequences of poses, which we used to extract trigger signals. Additionally, we used the distance, or how close the current posture is to the stored poses, to continuously morph between different sounds or timbres.

For some of the poses the quality of the suit calibration [9] could, to some degree, affect the resulting classification. We used a maximum of 9 different stored poses at one time. Furthermore, the recognition rate would probably decrease if we increased the amount of used poses. However, with a well selected set of pose features, it should be possible to use an extensive set of poses.

Feature	Used to control
Vertical position	Extensively for cont. and cond.
Relative positions	Trigger samples and cont.
Velocity (mag)	For cont., good "effort" relationship
Acceleration	Trig. sounds and state changes
QoM	Difficult for the performer to use
Relative body pos.	For cont. excitation and modification
Joint angles	Mostly for cond., some cont.
Poses	Notes, chords and states triggers

Table 1: Summary of how we used the different extracted features. There are three main uses of features, (1) continuous excitation o modification (cont.), (2) thresholded for use as trigger signals (trig.) and (3) as conditions for other triggers (cond.).

3. CONTROLLING SOUND AND MUSICAL FEATURES

3.1 The sound engine

All the sounds for the performance were generated and manipulated in Ableton Live 8 via MIDI and Open Sound Control (OSC). Ableton Live 8 does not accept OSC messages, so a third-party extension called LiveOSC was used to handle OSC data. However, we experienced considerable latency with the OSC messages, so time-critical events like synth notes, sound clips, and effects manipulation, had to be operated via MIDI.

The performance was organized in *states*, each containing

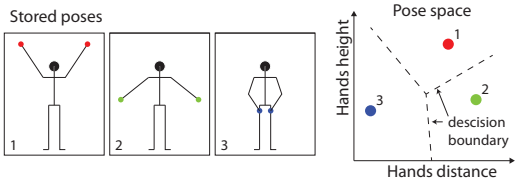


Figure 4: A simplified two-dimensional illustration of the pose classifier. The two pose features *hand height* and *hand distance* spans the pose space (right plot). Every pose will have a corresponding point in the pose space. We classify a pose to the one from the stored set that are closest.

sound effects, synths and other sound generating devices. As the performance progressed, we moved sequentially from one state to the next. A state could have various internal operations that affected Ableton Live 8, such as muting, raising volume, altering tempo, playing a clip, and so on. In the following Section we present how the states were controlled.

3.2 Transition between states

Our initial idea was to make a full-length performance piece in which all aspects of the performance were controlled solely by the Xsens MVN suit. For us, this meant that the performer needed to be, as much as possible, in full control of the whole performance. Therefore we needed to get rid of the invisible control center or the typical "guy behind the laptop"-setting [8].

At the same time we wanted the performance to have some varied content. We soon discovered that it was challenging to design a single instrument, or one synthesizer state, that would be interesting enough to listen to and watch for a whole performance. The performer needed to be able to change between different mappings. Our solution was to implement a so-called *finite-state machine*. This is a mathematical abstraction used to design sequential computer logic, which consists of a finite set of states, transitions between these states, and conditions for when the transitions should occur. To be able to go from one state to another the performer needed to perform predefined transition actions. Hence, the performer starts in one state, and when he/she feels that the part is finished, he/she can trigger the transition to the next state.

4. DISCUSSION

In the following we briefly discuss some of the thoughts we have had during the implementation of the Dance Jockey system.

4.1 Composing Dance Jockey

A challenge with composing and choreographing a performance for the Xsens MVN system was to decide to what degree the performance should be a musical concert controlled by a full body MoCap system, or a sonification of a dance piece [1]. We ended up with something in between. Designing action-sound mappings and making a performance around them turned the whole process into a creative one.

We also had to find a way to balance composition with improvisation. Some parts needed to be specified in detail, while others were left open. Specifically, parts featuring continuous sound excitation were particularly suitable for improvisation, and we found them to be especially important for establishing "expressive" action-sound relationships. The difference between a good and a bad concert was for us mostly determined by whether the performer was able to use these expressive parts to communicate with the audience.

4.2 The gap of execution

The process of composing and investigating action-sound mappings with the Xsens MVN suit takes a lot of time and energy. The suit is fairly quick to put on, but it is not comfortable to wear for several hours. It also involves many tiresome details, like calibration routines and changing batteries. While we were fully capable of performing concerts with the equipment, the time-consuming details and the obtrusiveness of the suit makes it tiresome to practice, compose and be creative.

Efficient tools are essential when attempting to compose and practice performances that employ full body MoCap technology. Through developing own tools and software while working with performance-related and technical aspects of the system, we have decreased the so-called gap of execution, or the gap between an idea - and its realization. Overcoming most of the technical challenges now enables us to focus on the artistic process. In this way our continued work on the Xsens performance will not be strangled by the many burdensome practicalities and obstacles that this technology and setup easily evokes.

4.3 Future research

We have seen a great number of possibilities that the Xsens MVN system offers for musical interaction, and feel that we have only touched the surface of these possibilities. Therefore, in the future we hope to get time and resources to make more thoroughly produced performances. We are currently working with more advanced action-sound mappings using physical models and granular synthesis, in order to build stronger perceptual connections between the MoCap data and sound output.

We also need to base our progression on more formal feedback. Up to now we have based our impressions on the feedback from audience members after concerts. This has not been sufficient to answer the questions we wanted to address, like: "Could you follow the action-sound mappings?" or "Did you enjoy the action-sound couplings or were they too evident/boring?" For that reason, in the future we would like to hand out questionnaires (likert scale, open ended questions, etc.) to get more formal feedback.

5. REFERENCES

[1] C. Bahn, T. Hahn, and D. Trueman. Physicality and feedback: a focus on the body in the performance of electronic music. In *Proc. of ICMC*, 2001.

[2] F. Bevilacqua, J. Ridenou, and D. Cuccia. 3D motion capture data: motion analysis and mapping to music. In *SIMS*, 2002.

[3] C. Dobrian. Aesthetic considerations in the use of 'virtual' music instruments. In *Proc. Workshop on Current Research Directions in Computer Music*, 2001.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[5] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. 2002.

[6] A. R. Jensenius. *ACTION - SOUND, Developing Methods and Tools to Study Music-Related Body Movement*. PhD thesis, University of Oslo, 2007.

[7] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman. Musical gestures: concepts and methods in research. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*, pages 12–35. Routledge, New York, 2010.

[8] M. Kimura. Creative process and performance practice of interactive computer music: a performer's tale. *Org. Sound*, 8:289–296, December 2003.

[9] S. A. Skogstad, K. Nymoen, Y. de Quay, and A. R. Jensenius. OSC Implementation and Evaluation of the Xsens MVN suit. In *Proc of NIME*, Oslo, Norway, 2011.

[10] S. A. Skogstad, K. Nymoen, and M. Hovin. Comparing inertial and optical mocap technologies for synthesis control. In *Proc. SMC*, 2011.

Paper V

Digital IIR Filters With Minimal Group Delay for Real-Time Applications.
S.A. Skogstad, S. Holm and M.E. Høvin.
In *IEEE The International Conference on Engineering and Technology. 2012.*, pages 1–6, German University in Cairo 2012.

Paper VI

Designing Digital IIR Low-Pass Differentiators With Multi-Objective Optimization.
S.A. Skogstad, S. Holm and M.E. Høvin.
In *IEEE 11th International Conference on Signal Processing. 2012.*, pages 10–15, Beijing Jiaotong University 2012.

Paper VII

Filtering Motion Capture Data for Real-Time Applications.
S.A. Skogstad, K. Nymoen, S. Holm, M.E. Høvin and A.R. Jensenius.
In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 196–197, Kaist University, Daejeon 2013.

Filtering Motion Capture Data for Real-Time Applications

Ståle A. Skogstad, Kristian Nymoen, Mats Hovin and Sverre Holm
University of Oslo, Department of Informatics
{savskogs,krisny,matsh,sverre}@ifi.uio.no

Alexander Refsum Jensenius
University of Oslo, Department of Musicology
a.r.jensenius@imv.uio.no

ABSTRACT

In this paper we present some custom designed filters for real-time motion capture applications. Our target application is *motion controllers*, i.e. systems that interpret hand motion for musical interaction. In earlier research we found effective methods to design nearly optimal filters for real-time applications. However, to be able to design suitable filters for our target application, it is necessary to establish the typical frequency content of the motion capture data we want to filter. This will again allow us to determine a reasonable *cutoff frequency* for the filters. We have therefore conducted an experiment in which we recorded the hand motion of 20 subjects. The frequency spectra of these data together with a method similar to the *residual analysis* method were then used to determine reasonable cutoff frequencies. Based on this experiment, we propose three cutoff frequencies for different scenarios and filtering needs: 5, 10 and 15 Hz, which correspond to *heavy*, *medium* and *light* filtering, respectively. Finally, we propose a range of *real-time* filters applicable to motion controllers. In particular, *low-pass filters* and *low-pass differentiators* of degrees one and two, which in our experience are the most useful filters for our target application.

1. INTRODUCTION

Motion capture (MoCap) and sensor technologies are often used for real-time interactive musical applications, e.g. game controllers like Wii Remote, PlayStation Move, Kinect, and other controllers like mobile phones and novel interfaces for desktop computers. The increased availability of new and improved MoCap technologies together with algorithms that interpret user motion as control data, make it increasingly affordable and feasible to use it for musical interaction. We refer to such interfaces as *motion controllers* (also known as *gesture controllers*) [6]. However, many MoCap and sensor technologies give noisy results, therefore making it necessary to apply noise removal filters [13, 18].

Low latency is a prerequisite for achieving intimate control in musical interactive applications [15]. And, as one might expect, there will always be a corresponding delay penalty when employing a *digital filter*. More specifically, this delay performance is given as the *group delay* and is measured in samples, or sampling periods. This further implies that the given time delay of a filter is proportional to the sampling rate of the MoCap system in use [7]. Since most MoCap systems have a relatively low sampling rate, normally between 30 and 200 Hz, this implies that the given group delay of the filter is critical for the total amount of delay. The goal of the current paper has been to develop filters that are optimized for motion controllers and that also minimize the latency they add to the musical applications (Figure 1).

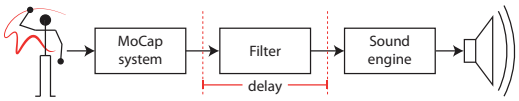


Figure 1: There is an intrinsic delay penalty when employing digital filters on MoCap data.

In our previous work we found methods to design nearly optimal *digital filters* with low group delay [11]. However, to be able to design application specific filters, it is necessary to determine the frequency properties of the data to be filtered. We have therefore conducted an experiment to determine these properties for musical application based on *free-hand motion in the air*.

In the next section we give a brief introduction to digital filters. Then, in section 3, we present the experiment and how to determine reasonable frequency properties of human MoCap data. Based on these results, a range of nearly optimal filters for the target application is presented, together with some evaluations in section 4, before the results are discussed in section 5.

2. BACKGROUND - DIGITAL FILTERS

Our main goal when applying *filters* is to smooth data or to restore signals that have been distorted with noise. There exist several methods, and they can roughly be divided into two categories; *curve fitting techniques* and *digital filters*. Curve fitting can intuitively be explained as trying to graphically fit a smooth curve to noisy data. The most common methods are *polynomial fit* and *spline methods* [18]. However, curve fitting noisy MoCap data is known to be suboptimal since human motion does not follow polynomial curves [9]. Digital filters are seen as the most general method for noise smoothing and is the technique we are going to adapt in this paper, since we want a *causal filter* with good real-time properties. Causal here indicates that the filter output depends only on past and present inputs, i.e. a mandatory property for real-time applications.

2.1 The filter objectives

Formally, the goal of a noise filter is to extract the desired signal from some noisy data. Typically this is done by designing a filter, with the purpose of removing the noise com-

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

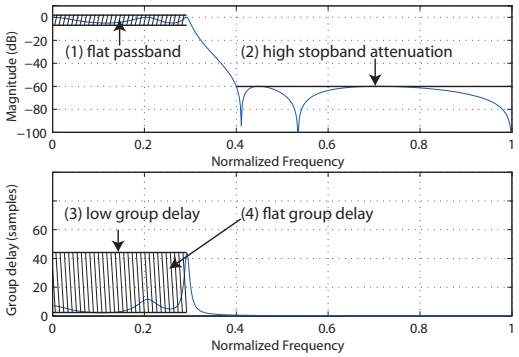


Figure 2: The frequency domain plot of an IIR low-pass filter. The filter objectives are highlighted.

ponent while leaving the desired signal unchanged. In other words, the main two filter objectives are:

- *Maximize noise attenuation.* That is, reduce the amount of noise to maximize the *signal-to-noise ratio* (SNR).
- *Minimize the signal distortion.* That is, avoid altering the desired signal.

There exists much theory regarding the two objectives above [17]. However, in this paper we are especially interested in the following additional objective:

- *Minimize the filter delay.* That is, to minimize the time it takes for the signal to pass the filter.

The most common way to design a digital filter is in the *frequency domain* [17]. Here the aim is to determining the localization of the signal and the noise in the frequency domain, and then designing an appropriate filter based on these properties. The *passband* refers to the frequencies that are passed, i.e. wanted, while the *stopband* refers to the frequencies we want to filter out. This technique works particularly well if the signal and the noise can be effectively separated in the frequency domain. However, this is not necessarily the case for MoCap data. For instance, so-called *white noise* is a common property for sensors [16], and is evenly distributed in the whole frequency band. In other words, not even an ideal low pass filter can suppress all the noise since there will also be noise in the passband [18]. In these cases we need to compromise between noise attenuation and signal distortion. We return to this challenge in section 3.

In Figure 2 we have plotted the frequency properties of a typical *low-pass* filter, which is the type we are going to work with since human motion mainly consists of low frequencies [18]. The figure highlights also the objectives of filter design. Simultaneously, we want: (1) *flat passband*, i.e. low signal distortion, (2) *high stopband attenuation*, i.e. high noise suppression, (3) *low group delay*, i.e. low latency, and (4) *flat group delay*, i.e. that all frequency components of the wanted signal are similarly delayed, also known as linear phase [7]. Let us now consider the different digital filter types.

2.2 Digital filter types (FIR and IIR)

There exist two main digital filter types, *finite impulse response* (FIR) filters and *infinite impulse response* (IIR) filters. *Moving average* is probably the most simple and intuitive realization of a FIR filter [14]. While the moving average filter have low-pass filter properties, the frequency domain properties are solely specified by it's length, i.e. the order of the filter. In most cases there will exist more optimal FIR filter solutions [14], but moving average filters are frequently used because they are intuitive and simple to implement.

IIR filters, as the name suggests, have an infinite impulse response that is the result of their recursive nature. While a FIR filter only bases its output on the input signal, an IIR filter bases its output on former output values as well. In essence, IIR filters offer an effective way of achieving a long *impulse response*, without having to use long FIR filters. Therefore, if the goal is to minimize the group delay, the use of IIR filters seems reasonable, since they can have dramatically lower order than symmetric FIR filters with similar performance [7]. Our results in [11] support this claim as well.

There is one main advantage to so-called *symmetric* FIR filters compared to causal IIR filters, being that they have *linear phase* which implies a constant group delay [17], i.e. all frequencies are delayed by the same amount. Symmetric FIR filters have additionally a fixed group delay of $n/2$ samples where n is the given filter order. In other words, their constant group delay comes at the expense of a fairly high filter delay compared to IIR filters with similar performance [11]. Furthermore, it is not certain that an IIR filter with a moderate amount of *group delay error* is a big concern for our target applications.

2.3 Low-pass differentiators (LPD)

Differentiators are a filter type that are commonly used to extract velocity and acceleration data from position data [13]. When differentiating MoCap data, it is normal to experience an increase of noise in the differentiated data. This is due to the fact that differentiation acts as a high pass filter. Accordingly, the low frequency motion data in the passband will be attenuated while the white noise in the higher frequencies will be amplified. As a result, we end up with a lower SNR value for the differentiated data, which increases the need for filtering [18, 2]. This is why it is reasonable to use so-called *low-pass differentiators*, since they avoid the undesirable amplification of noise in the higher frequency band. They also provide better total filter solutions than to use a low-pass filter in cascade with a differentiator operator, as we have shown in [10]. Similarly, it is better to use one low-pass differentiator of degree two, than to use two of degree one in cascade

2.4 Filter design methods

The design of symmetric FIR filters is a linear problem and there exist different general solutions for most FIR design problems, e.g. the *least square method* and the *Parks-McClellan* method [8, 4]. The design of IIR filters is, on the other hand, a nonlinear problem, and there are no general optimal design methods. There are however different construction methods, which can give optimal solutions for some special cases. The most known classical IIR filter methods are Butterworth, Chebychev and elliptical (Cauer) [17]. They are very useful for standard filter types as long as there is little restriction on the group delay responses [5, 11]. It is therefore necessary to use alternative design methods if we need more control over the group delay specifications. In our earlier research we presented a successful method for designing nearly optimal IIR filters with arbitrary specifications, including low-pass filters with minimal group delay [11] and IIR low-pass differentiators [10]. In that work we regarded filter design as a multi-objective optimization problem, which was solved using an unbiased metaheuristic

search algorithm. Using this method we are able to custom design nearly optimal IIR filters with the desired trade-off between group delay and the other filter objectives given above. For more details about this method see [10] and [11]. However, before we can design filters for our applications, we need to determine the typical frequency properties of the MoCap data we want to filter.

3. FREQUENCY PROPERTIES OF MOTION

As we show below, it is possible to determine reasonable cutoff frequencies from recorded MoCap data. The best method would be to determine the cutoff frequency before filtering a given set of data. However, this is impossible for real-time applications since the cutoff frequency needs to be specified beforehand. In practice, we are forced to use predetermined filters, and therefore need to estimate generic frequency properties for free-hand motion. Let us start by presenting our analysis methods before we continue with presenting the experiment in section 3.2.

3.1 Analysis methods

Before we can begin the discussion on how to estimate a reasonable generic cutoff frequency, we need to make some assumptions about the noise distribution of the relevant MoCap technologies. There can be many sources of noise in a MoCap system: it can be sensor noise, wobbling markers, electrical interference, quantization noise and more, dependent on the MoCap system used [19]. As already mentioned, sensors are known to have white noise properties [16, 19]. Some MoCap technologies may have a different noise distribution. However, for simplicity, in this paper we assume that the MoCap system has a white noise distribution. Consequently, our goal is to attenuate as much as possible of the frequency band that is not part of the signal band. If it is mandatory not to distort signal, we need to choose a cutoff frequency that is just outside the signal band. However, if we need higher noise suppression than is possible with this conservative choice, we need to compromise signal distortion by lowering the cutoff frequency inside the signal band [18]. The determination of the optimal cutoff frequency will then be based on the noise attenuation needed and how much we can lower the cutoff frequency inside the signal band without distorting the desired signals too much. To be able to determine the latter, we used the following two methods.

3.1.1 Power spectral density (PSD) estimation

The most common method to determine the frequency content of a digital signal is to analyze the *frequency spectrum*, which can be derived in different ways with the Fourier transform. A non smoothed spectrum estimation with the *Periodogram*, a classic non-parametric technique, will normally be too noisy to clearly show the trend in the data [3]. We therefore ended up using the *Welch's method* with a *Hann* window of length 100 (sampling frequency of 100 Hz). This is a much used method which reduces the noise in the spectral density estimation in exchange for reduced resolution in the frequency domain. However, other spectrum estimators and windows will give similar results [3].

3.1.2 Residual analysis

While the above mentioned method offers a good basis for making a conservative determination of the passband edge, it does not necessarily provide us with a good basis to determine a reasonable cutoff frequency. For a more hands on approach, it is possible to visually inspect the MoCap data when filtered with different cutoff frequencies. We can then choose the cutoff that provides a good balance between noise reduction and signal distortion. A more systematic version of this technique is known as *residual analysis*, which is a common method used for this task in the field of biomechanics [18]. The method consists of low-pass filtering the data with different cutoff frequencies and calculating the *residual*, i.e. what is left over when we subtract the filtered data from the raw data. As long as the filter is only attenuating noise, the residual should be rather small. However, when the filter starts to attenuate the desired signal, the residual will become larger. By performing this analysis for several cutoff frequencies, and plotting the resulting residuals, we get an overall picture of their impact. This plot can then serve as the basis for determining a reasonable cutoff frequency [18].

When computing the residual plots, care should be taken to make sure that the applied filters have constant group delay and are consistent with each other. This will ensure that the change in residual is not due to difference in the filter characteristics other than the cutoff frequency. It is common to use the actual intended filters which are supposed to be used in the final application [18]. However, our goal is not to find the optimal filter for a given set of data, but to find the main frequency trend of free-hand motion among several recordings. We ended up using the *window method* [7] to design the needed filters with an order of 200. This symmetric FIR design method has a broad cutoff frequency range and gives consistent filter characteristics for different cut-off frequencies [1].

3.2 The experiment

3.2.1 Setup and recordings

The experiment consisted of recording the hand motion of 20 subjects, 4 females and 16 males in the age range of 22-47. We used an optical infrared marker based MoCap system, OptiTrack, to record the subjects's hand motion at 100 Hz. The MoCap setup consisted of eight OptiTrack *V100:R2* cameras that were attached to tripods in a room measuring about 7x8 meters. One 16 mm reflective spherical marker was attached to the subject's dominant hand, close to the index finger, see Figure 3. Care was taken to minimize wobbling of the marker, which can introduce additional noise to the MoCap data. For the same reason, we also spent time calibrating the OptiTrack system. We did not want to perform post processing of the recorded data, e.g. for gap filling, which could potentially have distorted our results. Recordings with invalid or missing data were therefore omitted. The subject's hand motion were further recorded in the following two takes, both 20 seconds long.

- *Take 1:* The subjects were asked to move their dominant hand as rapid as possible in an arbitrary pattern. The intention of these recordings was to find an upper frequency limit for hand motion.
- *Take 2*: The subjects were asked to simulate that they were controlling some application with more *articulated* and *controlled* motion. Here we wanted to ex-

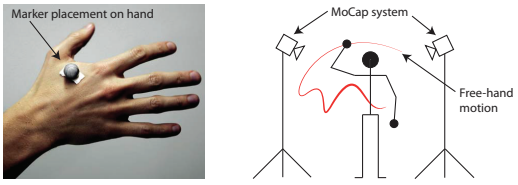


Figure 3: Placement of the marker (left) and an illustration of the experiment (right).

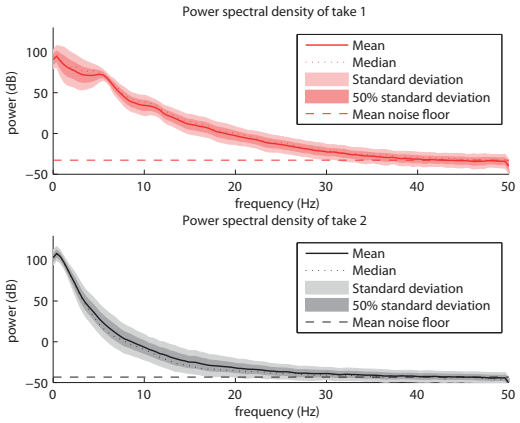


Figure 4: PSD estimation of the recorded data using Welch's method. The data is shown as statistical results of all 20 recordings, with results from both Take 1 (rapid) and Take 2.

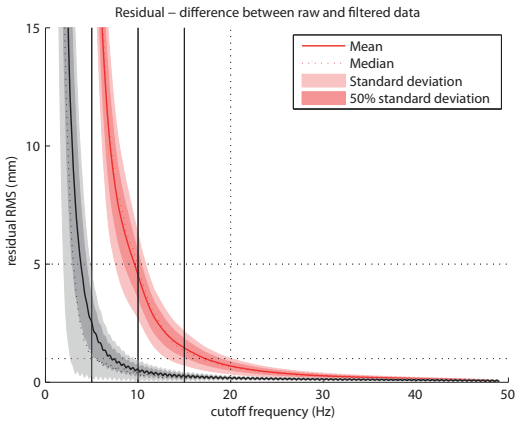


Figure 5: Statistical results of the *residual analysis* of the recorded data. Take 1 (rapid motion) is in red while take 2 is given in black.

amine the typical frequency content of the motion we anticipate to see most of in our target application.

We expected the latter to result in the need for a lower cutoff frequency than the former, which makes it possible to remove more noise. During all recordings, the subjects were asked to not clap their hands or make other limb collisions. We wanted to avoid collisions since they can be problematic to study, e.g. contain high frequency components that require higher sampling rates, and added noise problems with wobbling markers.

3.2.2 Results and interpretations

The results of the experiment are shown in Figures 4 and 5. As we can see from the spectral density estimates of Take 2, the mean value starts to move away from the noise floor between 20 and 30 Hz. For Take 1, the mean value starts to move away between 25 and 35 Hz. Furthermore, the main frequency content for Take 2 reaches roughly up to about 5–10 Hz, while Take 1 has a wider frequency distribution.

The residual plots in Figure 5 are somewhat easier to interpret since deviation in mm is more comprehensible than power in dB. When filtering hand motion, which normally has a displacement in the range of 200–1000 mm, a deviation of 1 mm is normally not significant. We have further seen a general trend for what the residual values indicates. When it was below 1 mm, the filters did not severely distort the MoCap data. But when the value increased above 5 - 10 mm, the filters started to clearly distort some high frequency parts of the MoCap data.

By using the above indicators and the statistical residual results in Figure 5, it seems reasonable to set the lower cutoff frequency for Take 2 to about 5 Hz, since the standard deviation is below 5 mm at this cutoff value. A reasonable upper frequency cutoff for Take 1, can further be set to be between 15 and 20 Hz, since the mean value goes below 1 mm in this region. A sensible trade off between these two outer cutoffs is in our opinion 10 Hz, since Take 2 is below 1 mm and Take 1 is below 5 mm for this cutoff value. Examples of how these cutoff frequencies perform can be seen in Figure 6. Based on this experiment, we propose the following three frequency cutoffs for filtering free-hand motion:

5 Hz *Heavy filtering*: Fast and rapid motion may be heavily smoothed out. However, the filtered data will contain the main features of normal controlled hand motion.

10 Hz *Medium filtering*: Most features of normal and medium rapid motion will be kept in the filtered data. However, some of the higher frequencies will be partially distorted.

15 Hz *Light filtering*: All main features of both rapid and normal motion are kept. Only the most extreme parts of the data may be partially blurred.

We could have added a cutoff frequency at 20 Hz, since the residual plot shows that the mean value of Take 1 decreases below 1 mm at about 20 Hz. But we have omitted this cutoff since we are not sure if the content that is blurred away with the 15 Hz cutoff, is due to noise or actual motion. The residual difference with the 20 Hz cutoff, is also minimal. However, a cutoff frequency of 20 Hz can be used if it is

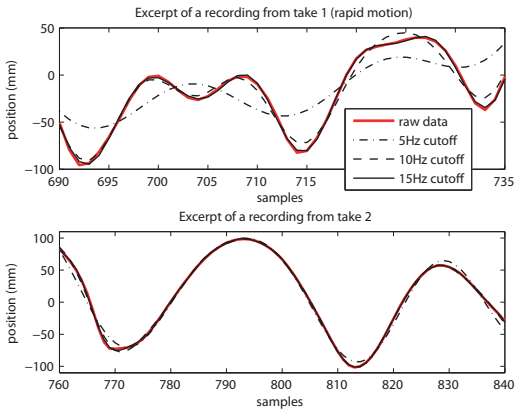


Figure 6: Excerpts from Take 1 and Take 2. While a 5 Hz filter cutoff works well for the Take 2 below, the rapid motion needs a 10 Hz or a 15 Hz cutoff frequency to follow the details in the recording.

important to keep all details in the recordings, and noise suppression is secondary.

3.2.3 Discussion

With this experiment we wanted to determine a generic trend in frequency content of free-hand motion. However, it was not straightforward to give instructions to the subjects. We hesitated to give them specific tasks, since this could lead them to do certain motion which could have influenced our results. We therefore ended up giving them quite general and open tasks, which resulted in a range of different interpretations and motion. However, as the results show, there is a quite clear trend among the recordings.

We considered testing expert subjects trained in moving at high frequencies, e.g. drummers. However, their motion is normally an effect of collisions and special techniques to be able to achieve high frequency. These motion were not part of our scope. Furthermore, inspection of the recorded data revealed that some contained *position jumps* that could not have been due to human motion. The errors clearly distorted the PSD data and raised the overall noise floor. It is therefore important to remove these errors if one wants valid PSD data. However, these errors had minimal impact on the residual plots, which shows that the residual method is a somewhat more robust analysis method.

4. PROPOSED IIR FILTERS

In our previous work we have based our *sound excitation* on three main types of MoCap data: *position*, *velocity* and *acceleration* [12]. We found these motion features to be the most useful for controlling sonic and musical features. We have therefore chosen to focus on the filter types that extracts these motion features from raw positional MoCap data, respectively *low-pass filters* and *low-pass differentiators* of degree *1* and *2*.

4.1 Proposed IIR vs. symmetric FIR filters

We have already shown in our previous work that our IIR design method can produce better low delay filters than currently available methods [10, 11]. As we can see from Table 1 and Figure 8, the proposed IIR filters are significantly better than symmetric FIR filters if low delay and high noise attenuation are of priority, giving a potential noise suppression gain between 5-16 dB for the relevant filter types. The presented IIR filters have a group delay of 2 samples or less. This group delay amount was found to give a well balanced trade-off between the different filter objectives. For a more thorough low-delay comparison between different filter types, see [11]. The specification of the proposed IIR filters is given on our project web page together with a MAX/MSP implementation [1], and a subset of these filters is given in Table 2. (To convert *normalized frequency* to hertz, multiply by half the sample frequency.)

4.2 Filter evaluation

We have tested the proposed IIR filters and confirmed their performance in MAX/MSP. It is not trivial to evaluate the filters for general NIME use as it depends strongly on the end application. While some applications may want to minimize noise to get the most robust performance, some applications may benefit artistically from MoCap noise as it can add a desirable texture to the resulting sound synthesis. Over-smoothing, i.e. deliberately distorting the signal, can also be appropriate for some applications. However, it is important to use a cutoff frequency that satisfies the need for the given task, as the following example shows. By identifying high peaks in the acceleration data, we are able to detect

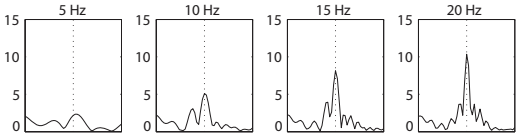


Figure 7: The effects of using different *cutoff frequencies* when extracting acceleration of a *hand clap*. The collision is more easily detected if the cutoff frequency is above 10 Hz (acceleration in m/s^2).

sudden motion and limb collisions, which we have used to trigger sonic and musical features [12]. The effect of using a too low cutoff frequency when extracting the acceleration data is shown in Figure 7. Not only does it attenuate more of the white noise, it also attenuates the acceleration peak. This is an expected effect, since a collision can be seen as an impulse which has a flat frequency response, i.e. the energy is spread out in the whole frequency band. The more of the frequency band that is included when differentiating, the more the collision power will be seen in the acceleration data.

Another important issue is what impact a moderate amount of *group delay error* can have on our target application. In our experience, there does not appear to be any dramatic negative distortion effect if the upper frequency range has some group delay error, as long as the main content (up to 5–10 Hz) has a fairly constant group delay. The optimized IIR filters are further superior if high noise attenuation, combined with low passband distortion and low group delay are desired. In our findings, it is possible achieve up to one-third the delay by using optimized IIR filters, as compared to symmetric FIR filters with similar performance. A delay of two samples, as opposed to six, yields a delay reduction of 40 ms for a MoCap system with a sampling frequency of 100 Hz, which should be a favorable reduction for a typical MoCap setup used for musical interaction [13]. In short, the optimized IIR filters have much better *low delay potential* than symmetric FIR filters for our target application, at the expense of a more complicated design.

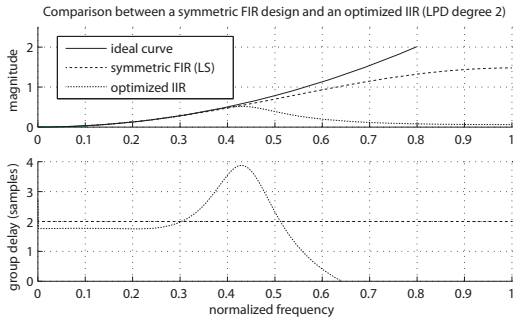


Figure 8: Comparison between 4th order low-pass differentiators (LPD) of degree 2 with a normalized cutoff frequency of 0.3. If low passband distortion is desired, the optimized IIR differentiator of degree 2, gives a noise suppression improvement of about 13 dB (≈4.5 times more noise attenuation) with similar or better performance for the other filter objectives.

Table 1: Potential noise attenuation gain in dB of the proposed IIR filters compared to optimal symmetric FIR designs, all of order 4. While the symmetric FIR filters have a fixed group delay of 2 samples, the proposed IIR filters have a group delay of 2 samples or less. For some of the proposed filters we have tolerated a moderate amount of group delay error.

normalized cutoff	0.1	0.2	0.3	0.4	0.5
low-pass filters	**8**	**8**	**8**	**6**	**5**
low-pass diff. of degree 1	**10**	**10**	**9**	**7**	**6**
low-pass diff. of degree 2	**16**	**15**	**13**	**12**	**10**

Table 2: Transfer functions for a subset of the proposed IIR filters with a maximum group delay of 2 samples and a normalized frequency cutoff of 0.2. The table presents a low-pass filter (LF) and low-pass differentiators of degree 1 and 2 (LPD1 and LPD2). More filters are presented on our project web page [1] .

type	$\frac{b_1}{a_1}$	$\frac{b_2}{a_2}$	$\frac{b_3}{b_3}$	$\frac{b_4}{b_4}$	$\frac{b_5}{b_5}$
LF	0.1227	-0.064575	0.044457	0.01949	0.019725
	1	-2.4965	2.8553	-1.5848	0.36631
LPD1	0.21077	-0.171566	-0.0552011	0.0182798	-0.00228255
	1	-1.71529	1.48777	-0.658224	0.118839
LPD2	-0.0797369	0.117985	-0.0144855	-0.00603732	-0.0177256
	1	-1.3405	1.19009	-0.531258	0.0931822

5. DISCUSSION AND CONCLUSION

In this paper we have addressed the challenge of using digital filters for real-time applications, focusing on filtering free-hand motion. To be able to design filters for such motion data, we conducted an experiment to determine the generic frequency properties of free-hand motion. Based on this experiment, we propose 3 different filter cutoffs; 5, 10 and 15 Hz. The 5 Hz, and partly the 10 Hz, cutoff will attenuate some of the high frequency parts of rapid free-hand motion. However, this may be necessary to get the needed noise suppression.

Although the experiment has only considered the frequency content of free-hand motion, our review of previous frequency studies in biomechanics suggests that most human motion is reported to be close to our found cutoff values, or more specifically between 3-26 Hz [9, 19, 20]. Our proposed frequency cutoffs should therefore work for most parts of the body, with some reasonable generalizations and adjustments, by regarding the kinematics of the used limb. Our proposed analysis method can be used if more certain knowledge is needed [1].

Finally, we propose a set of filters for our target applications, which has lower delay than what is achievable by established filter design methods. The main purpose of these filters has been to present some IIR filters designed with low group delay in mind, which is an important feature for intimate control for musical interactions. Compared to optimal symmetric FIR filters, they give a noise attenuation increase between 5-16 dB with similar delay, or up to 2-3 times the delay reduction for similar magnitude properties. These filters and some tools are published on our project page together with a Max/MSP implementation [1]. Since the optimal filter depends heavily on application specific details (e.g. sampling frequency, intended use), it is not possible to present a complete list of filters for all different applications and scenarios. However, our proposed set of filters should demonstrate the potential of using our filter design approach.

6. REFERENCES

[1] Project web page: fourms.uio.no/projects/sma/subprojects/mocapfilters.

[2] G. Giakas and V. Baltzopoulos. Optimal digital filtering requires a different cut-off frequency strategy for the determination of the higher derivatives. *Journal of biomechanics*, 30(8):851–855, 1997.

[3] M. Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, 1996.

[4] X. Lai. Optimal Design of Nonlinear-Phase FIR Filters With Prescribed Phase Error. *Signal Processing, IEEE Trans.*, (9):3399–3410, sept. 2009.

[5] M. Lang. Least-squares design of IIR filters with prescribed magnitude and phase responses and a pole radius constraint. *Signal Processing, IEEE Transactions on*, 48(11):3109 –3121, nov 2000.

[6] E. R. Miranda and M. Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, Inc., 2006.

[7] S. Mitra. *Digital signal processing: a computer based approach*. McGraw-Hill H. E., 2005.

[8] T. W. Parks and C. Burrus. *Digital filter design*. Topics in digital signal processing. Wiley, 1987.

[9] D. Robertson. *Research Methods in Biomechanics*. Human Kinetics, 2004.

[10] S. A. Skogstad, S. Holm, and M. Hovin. Designing Digital IIR Low-Pass Differentiators With Multi-Objective Optimization. pages 10 – 15. ICSP, IEEE Computer Society, 2012.

[11] S. A. Skogstad, S. Holm, and M. Hovin. Digital IIR Filters With Minimal Group Delay for Real-Time Applications. pages 1 – 6. ICET, IEEE Computer Society, 2012.

[12] S. A. Skogstad, K. Nymoen, Y. de Quay, and A. R. Jensenius. Developing the Dance Jockey system for musical interaction with the Xsens MVN suit. In *Proc. of NIME*, pages 226 – 229, Ann Arbor, Michigan, 2012.

[13] S. A. Skogstad, K. Nymoen, and M. Hovin. Comparing inertial and optical mocap technologies for synthesis control. In *Proc. Sound and Music Computing*, pages 421 – 426, Padova, 2011.

[14] S. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Tech. Pub., 1997.

[15] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *Proc. NIME*, Seattle, USA, 2001.

[16] J. S. Wilson. *Sensor technology handbook*, volume 1. Newnes, 2005.

[17] S. Winder. *Analog and digital filter design*. EDN series for design engineers. Newnes, 2002.

[18] D. Winter. *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons, 2009.

[19] G. Wood. Data smoothing and differentiation procedures in biomechanics. *Exercise and sport sciences reviews*, 10(1):308, 1982.

[20] J. Wosk and A. Voloshin. Wave attenuation in skeletons of young healthy persons. *Journal of Biomechanics*, 14(4):261–263, 1981.

Chapter 8

Appendix

8.1 The alternative filter design method

The alternative filter design method that has been used in this thesis is based on *multi-objective optimization* combined with a *heuristic search* method known as *random-restart hill climb*. The main idea behind this approach was to use an unbiased search method to be able to search freely for novel solutions without restricting the search space. More work could have gone into improving the optimization method. However, such discussion is out of the scope of this thesis. Our main requirement of unbiasedness was met, and the chosen algorithm has, to my experience, shown to be an effective algorithm for similar problems. Additionally, the resulting filters were also found to give credibility to the proposed method. In the following, I present how the filter design problem was formulated as a multi-objective optimization problem. Then I present the search algorithm and strategies that were used to solve these problems.

8.1.1 Multi-objective optimization (MOOP)

We can informally define *optimization* as the task of finding the *solution* that either maximizes or minimizes a problem. Since our design task consists of several objectives, given in Section 4.3.6, it is natural to regard it as a *multi-objective optimization problem* (MOOP) [12], which enables us to optimize several objectives simultaneously. This is done by combining the different objectives into one objective function. An important point with MOOP is that there will generally exist not a single optimal solution, but *several solutions* that depend on how we value the different objectives. Different weights w_i on the error functions err_i are used to specify how we value them. The weights, together with the error functions, then determine the search space, i.e. the function we want to optimize:

$$\text{minimize } Err = \sum_{i=1}^{4} w_i err_i \tag{8.1}$$

If we manage to create a search algorithm that can optimize this function for different weights, we can also determine the *noninferior surface* [10], i.e. the set of solutions that shows the best trade-off between the different objectives. We can then choose the solution on the surface that best suits our preference. These surfaces were essentially to make thorough com-

parisons between different design methods since they give a good image of the potential of the possible filters that the different methods can produce. This approach was used in Papers V and VI and Section 4.6 of this thesis.

8.1.2 Search strategy

Filter design parameters (search parameters)

When searching with classical filter design methods, the filter candidates were coded with the necessary parameters needed to determine the respective filter types. The unconstrained IIR search space was parameterized with pole and zero positions in the z-plane of the transfer functions. This made it simple to ensure stability by constraining the poles inside the unit circle while not constraining away possible optimal solutions [32]. It could also be used to achieve a stability margin by restricting the maximum radius of the poles. When searching after low-pass differentiators, one or two zeros were constrained to *dc* ($\omega = 0$) to get the wanted differentiator behavior (differentiators of degree 1 or 2, respectively). To evaluate the filter candidates, we used MATLAB to compute the error functions given in Equations (4.9), with a resolution of 100 uniformly spaced points in the frequency domain.

Search algorithm

To be able to search freely for new novel designs, we needed an unbiased optimization algorithm, i.e. one that makes little assumptions about the problem being optimized, also known as *metaheuristics*. *Random-restart hill climbing* was chosen as our search algorithm. This is an algorithm that combines the global view of *random search* with the local view of *hill climbing* [44]. Metaheuristics are not guaranteed to find optimal solutions. However, the chosen search algorithm was able to explore the search space satisfactorily if the exploration rate was chosen high enough compared with the search complexity. A pseudo-code of the used search algorithm, which shows the main behavior, is given below. The algorithm can be tuned in several ways, and there exist also alternative search algorithms. However, the discussion of these details is out of the scope of this thesis.

```
1  begin
2    repeat N times // N determines exploration rate
3      initialize random filter
4      climbedFilter = hillClimb(random filter)
5      if Err(climbedFilter) < Err(currentBestFilter)
6        currentBestFilter = climbedFilter
7    end
8  end
9
10 function hillClimb(filterCandiate)
11   initialize stepSize to 0.5
12   while (stepsize greater than 1E^8)
13     compute bestNeighboorFilter by adjusting each parameter...
14     of filterCandidate with +/-stepsize and check performance
15
16     if Err(bestNeigboorFilter) < Err(filterCandiate)
17       filterCandiate = bestNeigboorFilter
18     else
19       stepSize = stepsize/2
```

```
20    }
21    return filterCandiate
22  end
23
24  function Err(filterCoef)
25    compute err1, err2, err3 and err4 based on filterCoef (2)
26    return w1*err1 + w2*err2 + w3*err3 + w4*err4
27  end
```

Choosing the weights

The core problem with MOOP problems is finding the appropriate weights that give the wanted results. It is therefore important to find a weight strategy that makes it possible to uncover the wanted filters. In this thesis, three different weighting strategies have been used:

Adjusting w_3. In Paper V, we focused on minimizing the group delay while maximizing the stopband attenuation. We chose therefore *not* to incorporate the *group delay error* objective, i.e. w_4 was set to 0 for all the presented results. Furthermore, we mainly used the same fixed weights for objective functions err_1 and err_2 ($w_1 = w_2$). This was found to be a sensible balance and is also the same as what Cortelazzo et. al. used in [10]. As a result, we are left with only one weight that we need to adjust, the weight to our primary objective err_3 (low group delay). Thus, noninferior surfaces consisting of single lines can now be revealed by ramping w_3 from 0 to an appropriate value. A reasonable step size for this ramp was found by experimentation to achieve a somewhat even distribution of points in the noninferior surfaces.

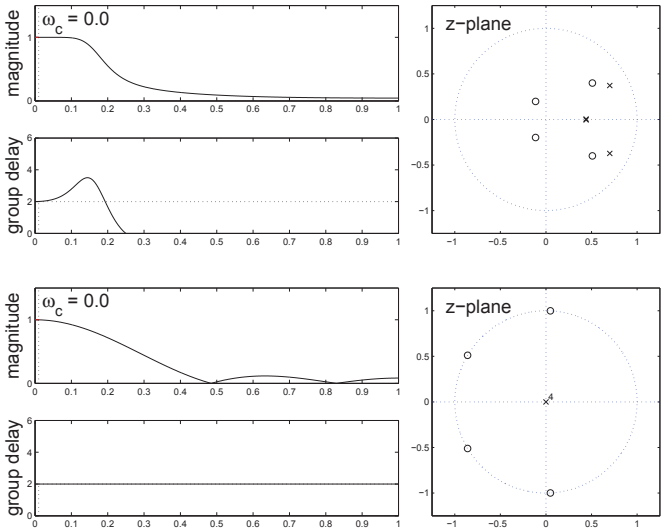
Adjusting the weight ratio between w_1 and w_2. In Paper VI, we focused on minimizing the passband error while maximizing the stopband attenuation. In other words, we wanted to find *magnitude optimal* filters. We choose therefore not to incorporate the group delay objective weights; i.e. w_3 and w_4 were set to 0 for all the presented results. This left us with two weights, w_1 and w_2. Finding the trade-off relationship between these two objectives was then just a matter of iteratively changing the weight ratio between w_1 and w_2 in order to reveal the noninferior surface.

Finding filters with a specific group delay. Section 4.6.2 gives a low-delay comparison between different filters with an upper group delay restriction of 2 samples. To be able to find a range of such filters, it was necessary to find a combination of weights w_1, w_2 and w_3 that gave different combinations of passband error and stopband attenuation, but with an upper group delay of two samples. These filters were found by employing a search algorithm that found the right combination of weights. More specifically, the filters were found by using different weight ratios between w_1 and w_2 and finding the corresponding weight w_3 that gave filters with an upper group delay of two samples.

An important difference between the above presented approach and the typically iterative optimization methods mentioned in Section 4.4.4 is that the above approach was not based on prescribing a specific constant group delay value. The wanted group delay response was found by finding the correct weights, which resulted in the wanted properties.

8.2 Reducing random noise filters

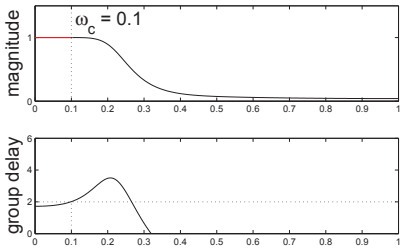
The specifications of the given filters from Section 4.7.4 are given below.

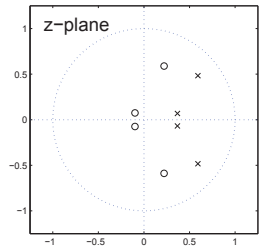


8.3 Proposed filters

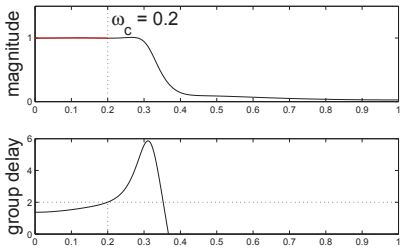
In Paper VII, we proposed a range of filters applicable for real-time MoCap applications, both low-pass filters and low-pass differentiators of degrees 1 and 2. The specifications of these IIR filters are given below by the pole and zero placements in the z-plane. The MATLAB function *zp2tf* can be used to convert the specifications to transfer functions. All filters have a group delay of two samples or less and have better low-delay performance than what currently established filter design methods can create (a noise attenuation gain between 5 and 16 dB compared with comparable symmetric FIR filters or two to four times the delay savings). Notice that it is more optimal to use one low-pass differentiator of degree 2 instead of using two subsequent low-pass differentiators in cascade.

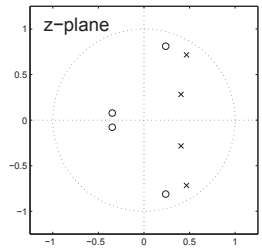
8.3.1 Low-pass filters



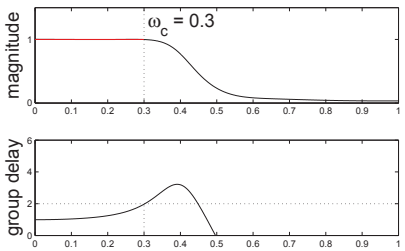


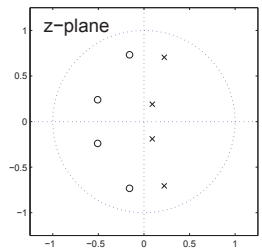

```
zeros =
 0.2212 + 0.5890i
 0.2212 - 0.5890i
-0.0985 + 0.0749i
-0.0985 - 0.0749i
poles =
 0.5920 + 0.4829i
 0.5920 - 0.4829i
 0.3672 + 0.0687i
 0.3672 - 0.0687i
gain = 0.1401
```



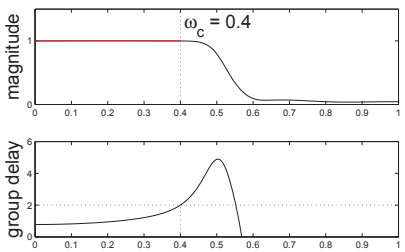


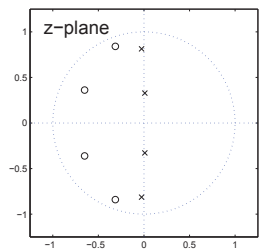

```
zeros =
 0.2390 + 0.8111i
 0.2390 - 0.8111i
-0.3482 + 0.0779i
-0.3482 - 0.0779i
poles =
 0.4669 + 0.7167i
 0.4669 - 0.7167i
 0.4062 + 0.2831i
 0.4062 - 0.2831i
gain = 0.1526
```



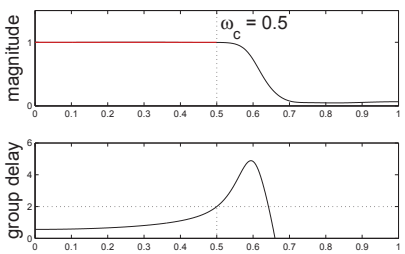


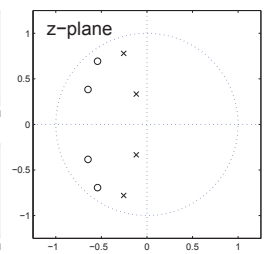

```
zeros =
-0.1581 + 0.7323i
-0.1581 - 0.7323i
-0.5082 + 0.2393i
-0.5082 - 0.2393i
poles =
 0.2229 + 0.7055i
 0.2229 - 0.7055i
 0.0913 + 0.1901i
 0.0913 - 0.1901i
gain = 0.2172
```





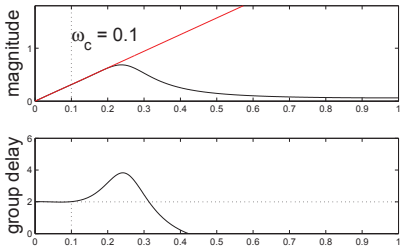

```
zeros =
-0.3137 + 0.8412i
-0.3137 - 0.8412i
-0.6514 + 0.3616i
-0.6514 - 0.3616i
poles =
-0.0258 + 0.8136i
-0.0258 - 0.8136i
 0.0096 + 0.3283i
 0.0096 - 0.3283i
gain = 0.2681
```

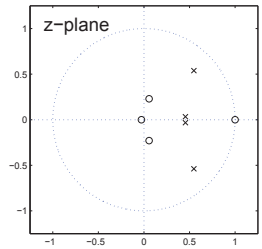




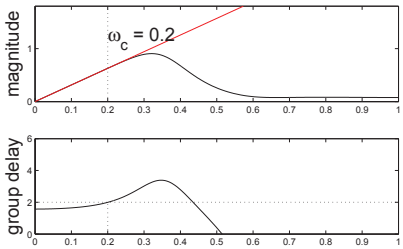

```
zeros =
-0.5421 + 0.6931i
-0.5421 - 0.6931i
-0.6468 + 0.3832i
-0.6468 - 0.3832i
poles =
-0.2549 + 0.7797i
-0.2549 - 0.7797i
-0.1173 + 0.3330i
-0.1173 - 0.3330i
gain = 0.3625
```

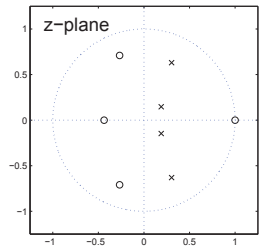
8.3.2 Low-pass differentiators of degree 1



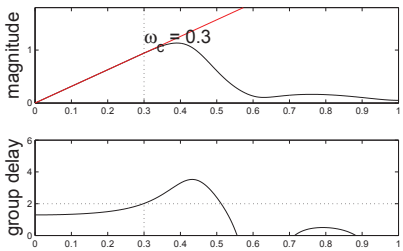


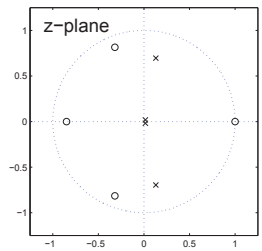

```
zeros =
 1.0000 - 0.0000i
 0.0562 + 0.2295i
 0.0562 - 0.2295i
-0.0274 - 0.0000i
poles =
 0.5479 + 0.5388i
 0.5479 - 0.5388i
 0.4555 + 0.0328i
 0.4555 - 0.0328i
gain = 0.1543
```



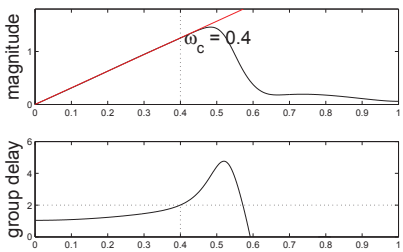


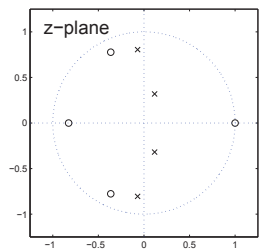

```
zeros =
 1.0000 - 0.0000i
-0.2671 + 0.7095i
-0.2671 - 0.7095i
-0.4371 - 0.0000i
poles =
 0.3038 + 0.6302i
 0.3038 - 0.6302i
 0.1898 + 0.1465i
 0.1898 - 0.1465i
gain = 0.1974
```



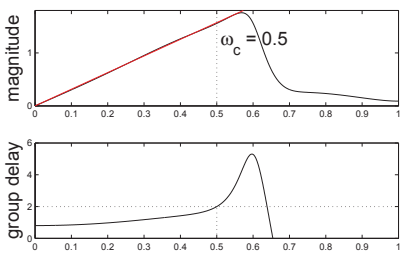


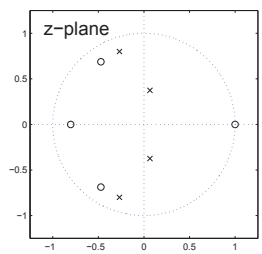

```
zeros =
 1.0000 - 0.0000i
-0.3191 + 0.8149i
-0.3191 - 0.8149i
-0.8497 - 0.0000i
poles =
 0.1306 + 0.6963i
 0.1306 - 0.6963i
 0.0154 + 0.0181i
 0.0154 - 0.0181i
gain = 0.2712
```





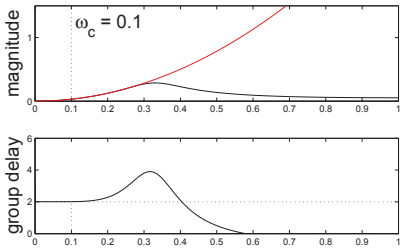

```
zeros =
 1.0000 - 0.0000i
-0.3641 + 0.7769i
-0.3641 - 0.7769i
-0.8263 - 0.0000i
poles =
-0.0695 + 0.8048i
-0.0695 - 0.8048i
 0.1167 + 0.3189i
 0.1167 - 0.3189i
gain = 0.3453
```

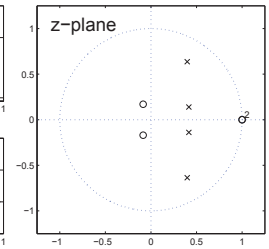




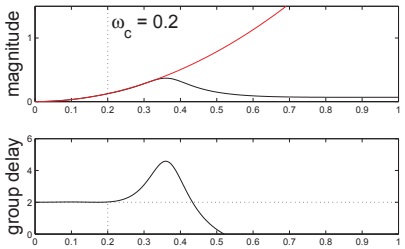

```
zeros =
 1.0000 - 0.0000i
-0.4720 + 0.6877i
-0.4720 - 0.6877i
-0.8034 - 0.0000i
poles =
-0.2689 + 0.8002i
-0.2689 - 0.8002i
 0.0662 + 0.3745i
 0.0662 - 0.3745i
gain = 0.4566
```

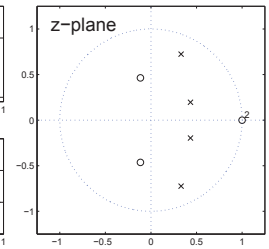
8.3.3 Low-pass differentiators of degree 2



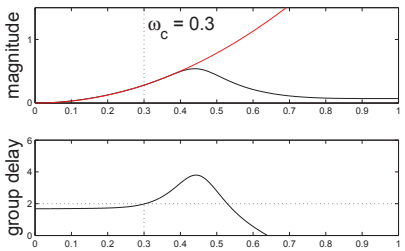


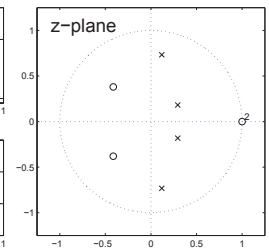

```
zeros =
 1.0000 + 0.0000i
 1.0000 - 0.0000i
-0.0855 + 0.1698i
-0.0855 - 0.1698i
poles =
 0.3981 + 0.6366i
 0.3981 - 0.6366i
 0.4160 + 0.1394i
 0.4160 - 0.1394i
gain = 0.0739
```



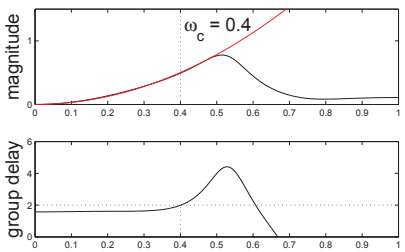


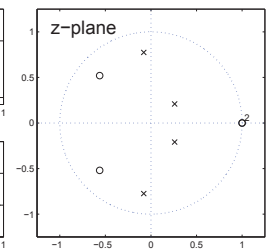

```
zeros =
 1.0000 - 0.0000i
 1.0000 - 0.0000i
-0.1156 + 0.4639i
-0.1156 - 0.4639i
poles =
 0.3304 + 0.7244i
 0.3304 - 0.7244i
 0.4335 + 0.1958i
 0.4335 - 0.1958i
gain = 0.0812
```



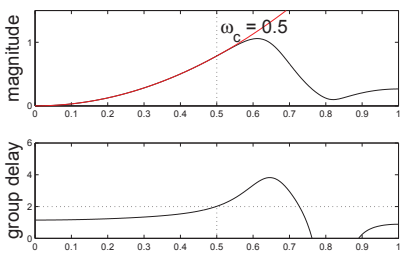


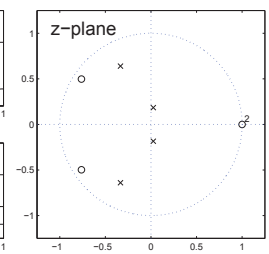

```
zeros =
 1.0000 - 0.0000i
 1.0000 - 0.0000i
-0.4136 + 0.3798i
-0.4136 - 0.3798i
poles =
 0.1192 + 0.7326i
 0.1192 - 0.7326i
 0.2946 + 0.1817i
 0.2946 - 0.1817i
gain = 0.1099
```






```
zeros =
 1.0000 + 0.0000i
 1.0000 - 0.0000i
-0.5620 + 0.5199i
-0.5620 - 0.5199i
poles =
-0.0789 + 0.7748i
-0.0789 - 0.7748i
 0.2605 + 0.2093i
 0.2605 - 0.2093i
gain = 0.1396
```






```
zeros =
-0.7629 + 0.4974i
-0.7629 - 0.4974i
 1.0000 - 0.0000i
 1.0000 - 0.0000i
poles =
-0.3339 + 0.6397i
-0.3339 - 0.6397i
 0.0260 + 0.1846i
 0.0260 - 0.1846i
gain = 0.2032
```